HEIMCOMPUTER SPIELEND LEICHT

Ein praktisches Computer-Handbuch

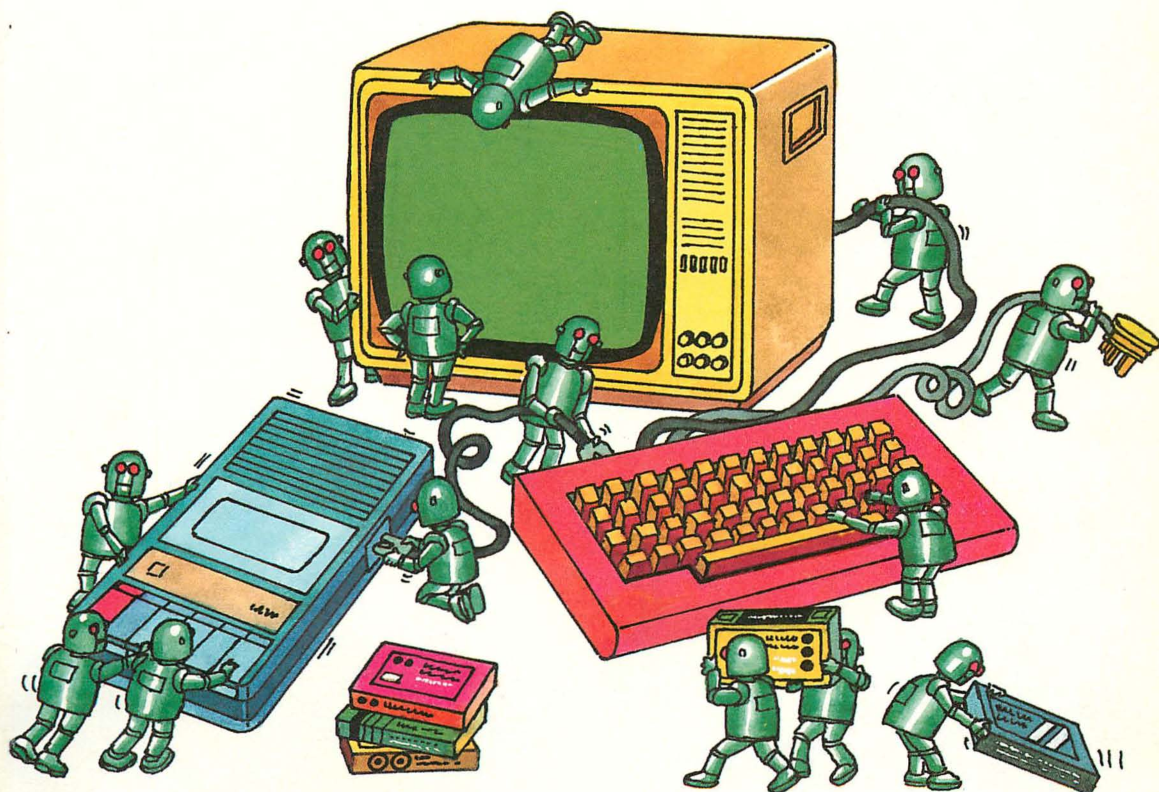


Bill Bennett · Brian Reffin Smith · Corinne Stockley
Judy Tatchell · Lisa Watts

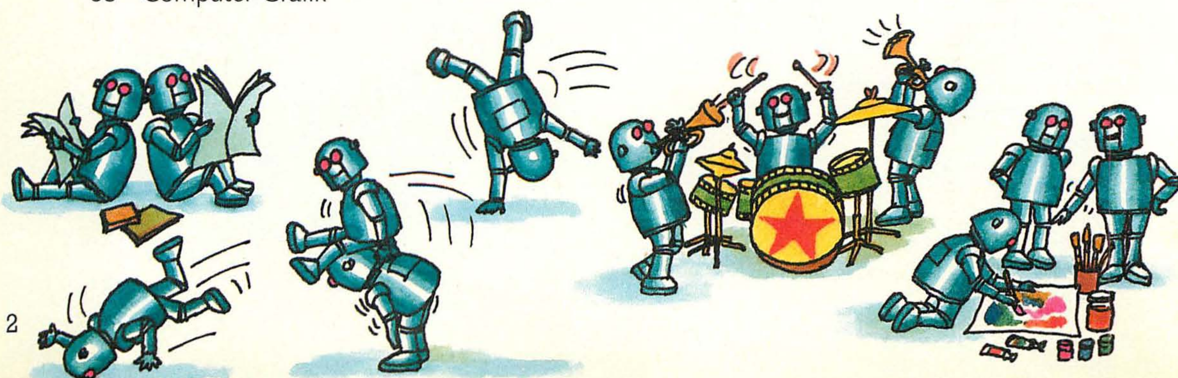
HEIMCOMPUTER
spielend leicht

Ein praktisches Computer-Handbuch für Einsteiger

Aus dem Englischen übersetzt von Barbara Schumacher

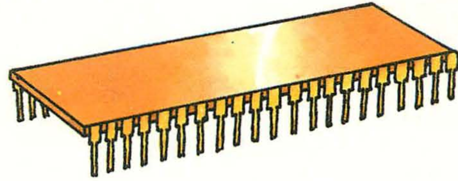


Inhalt

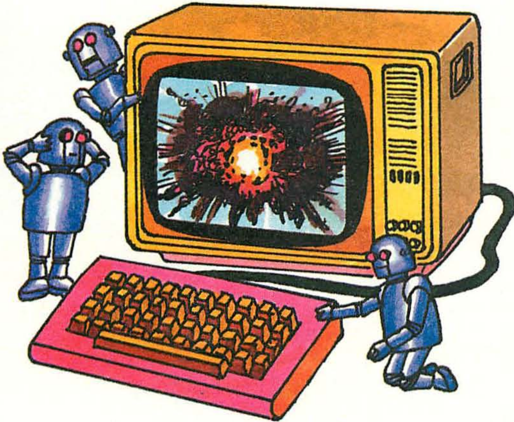


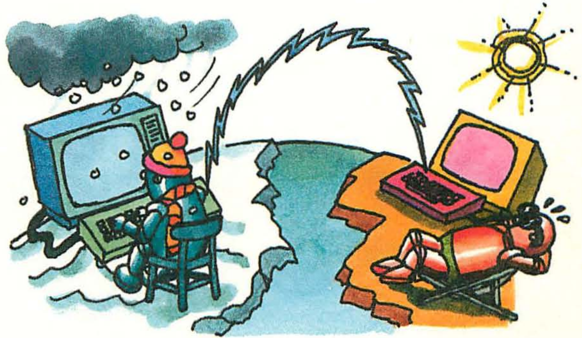
Über dieses Buch

Dieses Buch ist für alle geschrieben, die etwas über Mikrocomputer wissen möchten. Es zeigt, was man mit ihnen machen kann, wo sie sich einsetzen lassen und wie sie arbeiten. Es erklärt die wichtigsten Fachausdrücke, so daß man anschließend mehr über Computer lesen und verstehen kann.



Das Buch erklärt auch, wie ein Computer arbeitet und wie er Bilder und verschiedene Töne erzeugt. Es zeigt das Innere eines Mikrocomputers mit seinen winzigen Siliziumplättchen (Chips), die die gesamte Arbeit leisten.
Man sieht auch, wie man Computer miteinander verbindet, die Tausende von Kilometern voneinander entfernt stehen können, um die verschiedensten Arten von Informationen verfügbar zu machen. Mikrocomputer können auch dazu dienen, Roboter oder elektronisches Spielzeug zu steuern, wie etwa eine Modelleisenbahn oder eine Heizung.

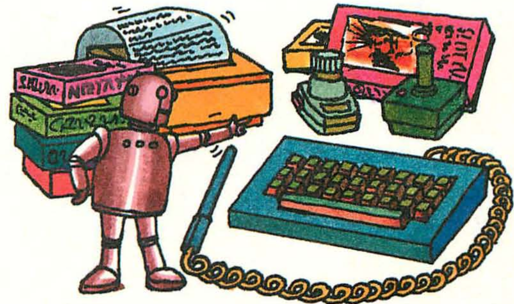




Mikrocomputer sind kleine, vielseitig einsetzbare Computer. Man kann mit ihnen spielen, malen, Töne erzeugen und sogar Musik machen. Sie können sehr schnell schwierige Rechnungen ausführen, und man kann mit ihrer Hilfe einen Terminkalender, eine Schallplattensammlung oder ein Fotoarchiv führen.



Für den Anfang genügt ein normales Fernsehgerät, das an den Mikrocomputer angeschlossen wird. Man kann später weitere Geräte hinzukaufen, die man mit dem Mikrocomputer verbindet, z. B. einen Lichtstift, mit dem man direkt auf dem Bildschirm zeichnet, oder spezielle Vorrichtungen für Computerspiele. Das Buch beschreibt auch diese Geräte.

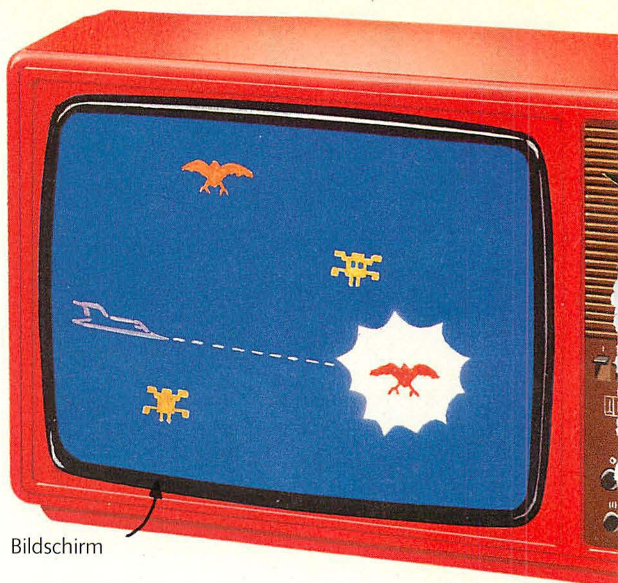


Der erste Teil des Buches erklärt, wie ein Mikrocomputer funktioniert und wie man ihn richtig bedient.
Der zweite Teil bietet eine Einführung in die Programmiersprache BASIC, die die meisten Mikrocomputer verstehen. Praktische Hinweise, Übungsbeispiele und einige Spiele erleichtern den Einstieg für den frischgebackenen Computer-Besitzer.

Im dritten Teil des Buches werden die Fachausdrücke erklärt, die hier gebraucht werden, um Computer, ihre Arbeitsweise und Bedienung zu beschreiben.

Das ist ein Mikrocomputer

Diese beiden Seiten zeigen, wie ein Mikrocomputer angeschlossen wird. Nicht alle Mikrocomputer sehen genauso aus wie dieser hier. Die meisten Heimcomputer bestehen jedoch aus einer Tastatur, die man an das Fernsehgerät anschließt. Einige Mikrocomputer haben spezielle Bildschirme oder Monitoren. Zu allen neuen Mikrocomputern gibt es Handbücher mit Gebrauchsanweisungen. Bevor man einen Mikrocomputer anschließt, sollte man das zugehörige Handbuch lesen.



Die Tastatur
Befehle und Daten werden mit Hilfe einer Tastatur eingetippt.

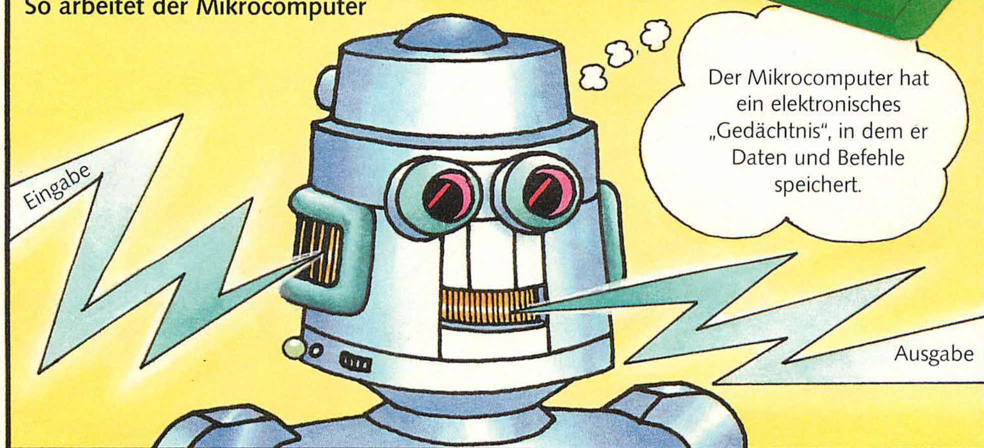
Gehäuse mit Tastatur

Anschlüsse für die Verbindungen zum Fernsehgerät und zum Stromnetz

Alle wichtigen Teile des Mikrocomputers befinden sich im Gehäuse. Dort wird die gesamte Arbeit verrichtet.



So arbeitet der Mikrocomputer



Das „Gehirn" des Mikrocomputers liegt im Gehäuse. Es besteht aus dem *zentralen Prozessorelement* (engl. **C**entral **P**rocessing **U**nit = CPU), das die gesamte Arbeit leistet, und einem Speicher. Bevor irgend etwas getan werden kann, braucht der Prozessor eine bestimmte Menge von Anweisungen. Diese nennt man *Programm*. Das Programm wird im „Gedächtnis" des Computers gespeichert, zusammen mit den Informationen oder Daten, mit denen der Computer arbeiten soll. Programme und Daten heißen *Eingabe*. Die Ergebnisse heißen *Ausgabe*.

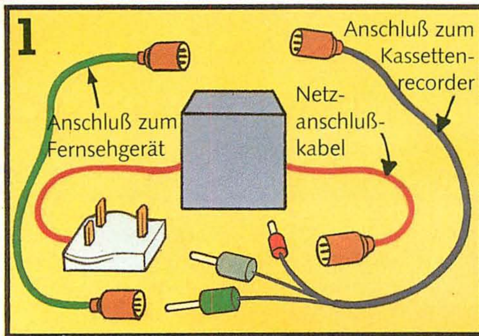




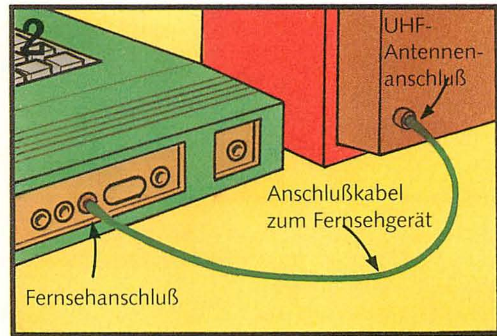
Der Bildschirm
Alles, was auf der Tastatur eingetippt wird, erscheint auf dem Bildschirm, ebenso wie die Ergebnisse der Arbeit des Mikrocomputers. Der Mikrocomputer kann auch Bilder malen und Figuren auf dem Bildschirm erscheinen lassen. An ein Farbfernsehgerät angeschlossen, können die meisten Mikrocomputer sogar farbige Bilder herstellen.

Das Netzteil
Das Netzteil transformiert die normale Netzspannung und paßt sie den Erfordernissen des Computers an.

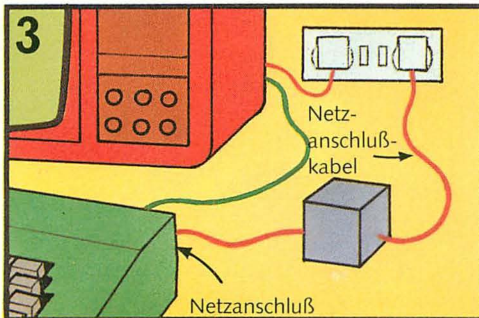
So schließt man einen Mikrocomputer an



Die meisten Mikrocomputer haben drei Kabel: eines zur Verbindung mit dem Fernsehgerät, eines für den Netzanschluß und ein drittes zur Verbindung mit einem Kassettenrecorder. Diesen benutzt man, um Programme für den Mikrocomputer zu speichern. Näheres darüber auf Seite 16.



Um die Tastatur mit dem Fernsehgerät zu verbinden, zieht man den Antennenstecker aus dem Fernsehgerät. Dann steckt man das eine Ende des Kabels in den mit TV gekennzeichneten Anschluß der Tastatur und das andere Ende in den UHF-Antennenanschluß des Fernsehgeräts.



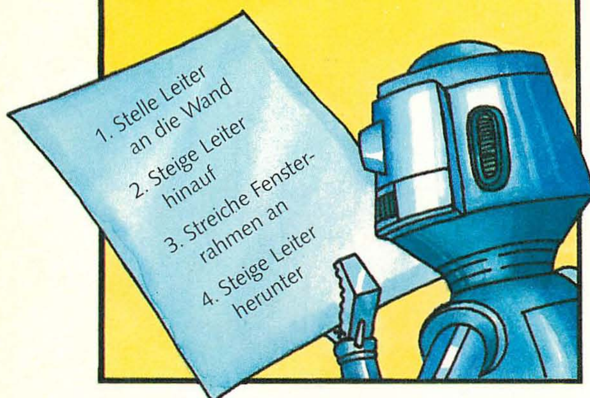
Man steckt ein Ende des Netzanschlußkabels in die Buchse der Tastatur, das andere Ende in die Steckdose. Hat man sich vergewissert, daß das Fernsehgerät angeschlossen ist, können beide Geräte eingeschaltet werden.



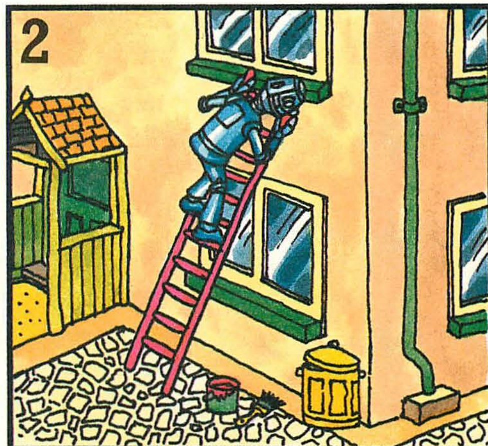
Man wählt einen Fernsehkanal, der nicht für das Fernsehprogramm benutzt wird, und stellt das Fernsehgerät ein, bis das Bereitzeichen des Mikrocomputers auf dem Bildschirm erscheint. Dieses Zeichen ist von Computer zu Computer verschieden.

Wie man einen Mikrocomputer programmiert

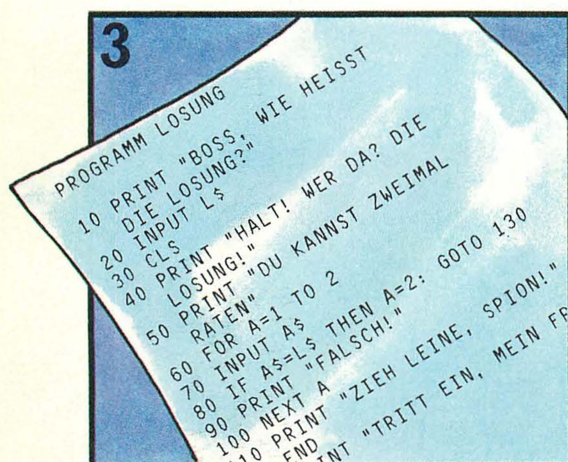
Ganz gleich, ob man den Mikrocomputer dazu benutzen will, ein Spiel zu spielen oder nur ein paar Zahlen zusammenzurechnen, man muß ihm auf jeden Fall ein Programm geben, das ihm sagt, was er zu tun hat. Dafür gibt es spezielle Computersprachen, in denen man Programme schreibt. Computersprachen bestehen aus Wörtern und Zeichen, die der Computer erkennen und in seinen eigenen elektronischen Code, den *Maschinencode*, umwandeln kann. Programmbefehle werden im Gedächtnis des Computers gespeichert und dann vom Prozessorelement ausgeführt. Die Programme und Daten, die man in den Computer eingibt, heißen *Software*. Die Teile des Mikrocomputers, die man anfassen kann, sind die *Hardware*.



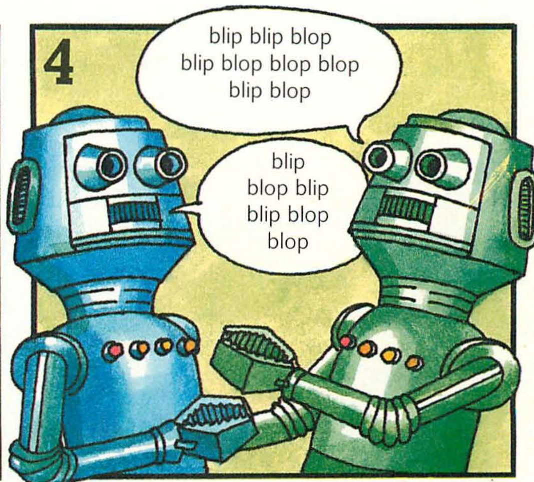
Ein Computer kann nur dann eine Aufgabe erfüllen, wenn man ihm genau und in der richtigen Reihenfolge sagt, was er tun soll.
Dieses Programm sagt einem Roboter mit Computergehirn, wie ein Fensterrahmen zu streichen ist.



Das Programm würde aber nicht funktionieren: Es fehlt nämlich die Anweisung an den Roboter, Pinsel und Farbtopf mitzunehmen, bevor er auf die Leiter steigt. Der Roboter tut nur das, was man ihm vorschreibt.



Dies ist Teil eines Programms in der Programmiersprache BASIC, die von den meisten Computern verstanden wird. Der Computer enthält einen „Übersetzer" (Interpreter), der die Programmiersprache in den Maschinencode übersetzt (siehe auch Seite 12).



Die gesamte Arbeit im Inneren des Computers wird mit Hilfe des Maschinencodes ausgeführt. Jedes „Wort" dieses Codes besteht aus Kombinationen von Impulsen des elektrischen Stroms, der im Computer fließt.

Das Gedächtnis des Computers – der Speicher

Ein Computer hat zwei Arten von Speichern. Der eine ist ein Dauerspeicher, der die Arbeitsbefehle enthält. Der andere Speicher ist ein leerer Speicher, in dem Programme und Daten für eine Aufgabe vorübergehend gespeichert werden. Immer, wenn der Mikrocomputer abgeschaltet wird, wird dieser Speicher wieder gelöscht.

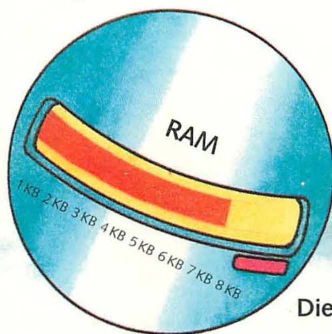
Der „zeitweilige" Speicher heißt RAM (engl. **R**andom **A**ccess **M**emory). Man nennt ihn auch *Schreib-/Lesespeicher*. Alles, was dem Mikrocomputer eingegeben wird, wird hier gespeichert oder „abgelegt", damit man es „lesen" oder darauf verweisen kann. Man kann den Speicherinhalt auch abändern.

Der „Dauerspeicher" heißt ROM (engl. **R**ead **O**nly **M**emory). Man nennt ihn auch *Festwertspeicher*. Der Name bedeutet, daß der Computer nur Informationen entnehmen oder „lesen" kann. Man kann in diesem Speicher keine Informationen speichern. Der Interpreter ist im ROM gespeichert.

▲ ROM ist wie ein Handbuch. Der Computer kann nur daraus lesen, er kann nichts „ausradieren" und keine neuen Informationen hinzufügen.

▲ RAM ist wie ein Notizbuch. Der Computer kann sowohl „hineinschreiben" als auch darin lesen. Sobald der Mikrocomputer abgeschaltet ist, wird der Inhalt dieses Speichers gelöscht.





Die Speicherfähigkeit

Es gibt Mikrocomputer mit unterschiedlicher Speicherfähigkeit. Die Größe des Speichers wird bestimmt durch die Anzahl der im Maschinencode geschriebenen „Wörter", die gespeichert werden können. Jedes Code-Wort ist ein *Byte*; 1024 Bytes ergeben ein *Kilobyte* (1 KB).

Ein Kilobyte entspricht etwa 500 BASIC-Wörtern oder Zeichen. Das genügt, um einfache Programme zu speichern. Kompliziertere Programme sind länger und brauchen 8 KB oder 16 KB RAM. Man kann bei den meisten Mikrocomputern den Speicherplatz erweitern.

Das ist die Tastatur

In der Regel entspricht die Tastatur eines Mikrocomputers der einer Schreibmaschine. Sie enthält die gleichen Buchstaben und Zahlen in derselben Anordnung. Eine Mikrocomputer-Tastatur hat jedoch noch einige zusätzliche Tasten für spezielle Befehle in der Programmiersprache BASIC. Der Mikrocomputer erhält von den einzelnen Tasten verschiedene elektrische Impulse. Wenn man etwas eintippt, was der Computer nicht versteht, dann erscheint das Wort ERROR oder FEHLER auf dem Monitor. Alles, was man eintippt, wird im zeitweiligen Speicher (RAM) des Computers gespeichert und erscheint auf dem Bildschirm, damit man es kontrollieren kann. Auf diesen beiden Seiten stellen wir zwei unterschiedliche Tastaturen vor.

Die Buchstabentasten
Bei vielen Computern kann man ein Programm eingeben, indem man die Zeichentasten benutzt und die Wörter mit Hilfe der Buchstabentasten buchstabiert.

Dies ist das Zeichen für „mal" (Multiplikationszeichen).

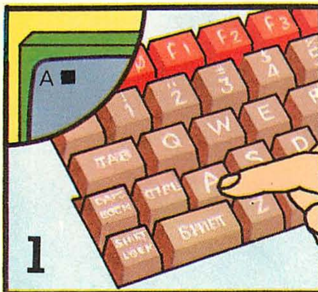
Umschalttaste und Feststelltaste für Großbuchstaben

Manche Mikrocomputer haben vielleicht einige andere Tasten.

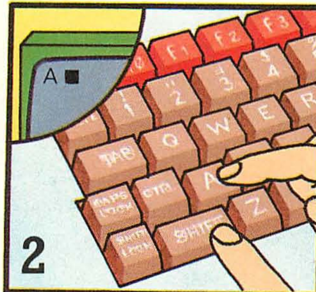


Die Leertaste
Diese Taste drückt man, um einen Zwischenraum zwischen Wörtern oder Zeichen zu setzen.

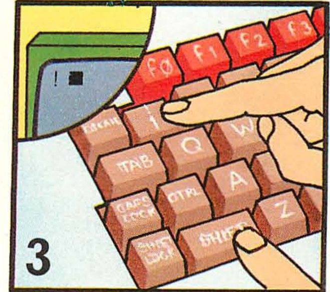
Die Umschalttaste



Die meisten Mikrocomputer schreiben automatisch Großbuchstaben auf den Monitor; sie können keine kleinen Buchstaben schreiben.



Einige Mikrocomputer können auch Kleinbuchstaben schreiben. Um Großbuchstaben zu erhalten, muß man zunächst die Umschalttaste drücken und dann die Buchstabentaste antippen.



Wenn eine Taste mit zwei Zeichen belegt ist, bekommt man das obere Zeichen, indem man die Umschalttaste drückt. Tippt man eine Taste an, ohne gleichzeitig die Umschalttaste zu drücken, dann erhält man das untere Zeichen.

Programmierbare Tasten
Dies sind spezielle Tasten, die man selbst für bestimmte Aufgaben programmieren kann, z. B. um bestimmte Farben zu erzeugen. Nicht alle Mikrocomputer haben diese Tasten.

Die Ziffer Null auf einer Mikrocomputer-Taste hat einen Schrägstrich, um sich von dem Großbuchstaben O zu unterscheiden.

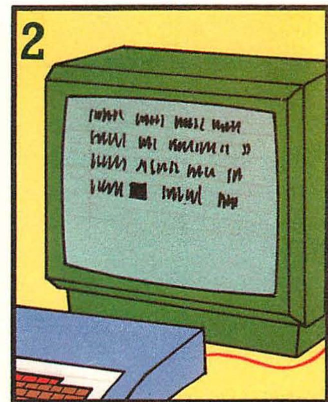
Cursor-Kontrolltasten

Die RETURN-Taste
Diese Taste drückt man am Ende jeder Programmzeile, damit man eine neue Zeile anfangen kann. Sie heißt manchmal auch NEWLINE oder ENTER.

Die Löschtaste
Mit dieser Taste können Fehler korrigiert werden.Sie heißt manchmal auch RUBOUT oder ERASE.



Der Cursor ist eine kleine Lichtmarke, die auf dem Bildschirm erscheint, während man tippt. Sie zeigt, wo der nächste Buchstabe erscheinen wird.



Wenn man etwas ändern oder löschen möchte, kann man den Cursor mit den Cursor-Kontrolltasten über das bereits Geschriebene zurückführen.

Ein anderer Mikrocomputer

Diese Tastatur ist nur ungefähr ein Viertel so groß wie die Tastatur auf der gegenüberliegenden Seite. Die Ausführung des Tastenfeldes kann Größe und Form eines Mikrocomputers bestimmen, denn die Bauteile im Inneren sind sehr klein. Normalerweise hat das Tastaturgehäuse auch Anschlüsse für Kabel zu anderen Geräten, z. B. zum Drucker oder zum Kassettenrecorder. Diese werden später beschrieben.



Dieser Mikrocomputer hat ähnliche Tasten wie ein Taschenrechner. Sie lassen sich nicht weit herunterdrücken. Die meisten Tasten enthalten vollständige BASIC-Wörter, so daß diese nicht Buchstabe für Buchstabe eingetippt werden müssen. Die Tasten sind mit Wörtern, Buchstaben und Zeichen belegt. Es gibt zwei verschiedene Umschalttasten für die Auswahl des gewünschten Wortes, Buchstabens oder Zeichens.

Programme für den Mikrocomputer

Man kann viele fertige Programme kaufen. Man findet sie in Zeitschriften und Büchern, oder sie sind auf Kassetten oder Disketten gespeichert. Man kann aber auch lernen, eigene Programme zu schreiben. Programme, die Zeile für Zeile ausgedruckt werden, heißen *Listing*. Kassettenprogramme können mit Hilfe eines Kassettenrecorders in einen Mikrocomputer eingegeben werden. Das Programm muß in der für den Computer richtigen Sprache geschrieben werden. Dies ist meistens BASIC; es gibt jedoch verschiedene BASIC-„Dialekte" mit unterschiedlichen Befehlen. Ein Programm, das Fehler hat oder nicht im richtigen Dialekt geschrieben ist, funktioniert nicht.



Man kann die verschiedensten Spiele kaufen – von Spielen mit farbigen Bildern und Klangeffekten bis zu traditionelleren Spielen wie Schach.

So kommt man zu Programmen

Die Mikrocomputer-Zeitschriften bieten Programmausdrucke an. Manche Zeitschriften behandeln nur bestimmte Computer. Andere enthalten Programme für verschiedene Computertypen.

Es gibt auch Bücher mit Programmen für viele Verwendungsmöglichkeiten (Spiele, Buchhaltung, Dateien usw.).

Leider enthalten die Programme, die man in Zeitschriften findet, manchmal Fehler, da sie nicht genügend getestet wurden. Programme in Büchern sind meist besser.

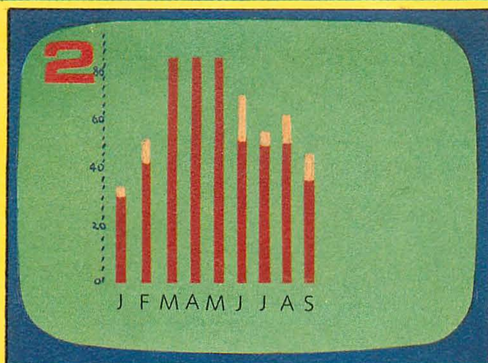
BASIC kann man gut üben, indem man Programme eingibt und sie an den Dialekt des eigenen Mikrocomputers anpaßt.

Bildschirmtext

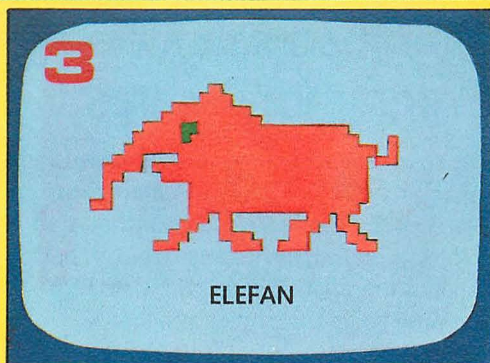
ROM-Kassetten

Manchmal gibt es auch Programme auf dem Bildschirm. Dazu braucht man ein Fernsehgerät mit Bildschirmtext-Anschluß. Bildschirmtext (BTX) kommt von einem elektronischen Informationszentrum. Man kann einzelne Informationsseiten auswählen, die dann auf dem Bildschirm erscheinen.

Für einige Mikrocomputer gibt es Programme auf Kassetten. Man schiebt die Kassette in den Mikrocomputer ein, und das Programm wird automatisch in den Speicher des Computers geladen.



Es gibt Programme für den privaten Bereich und für den Haushalt: zur Kontoführung, zum Führen eines Terminkalenders sowie zur Katalogisierung von Sammelobjekten. Dies sind einfache Ausführungen derjenigen Programme, die im Geschäftsleben benutzt werden.

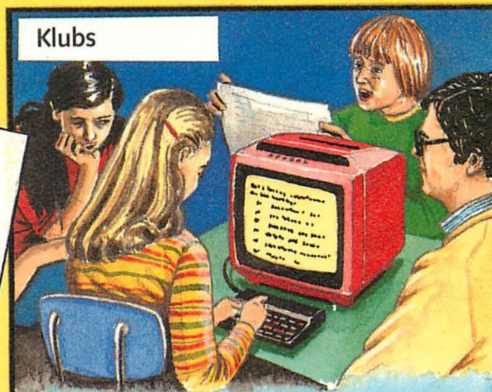


Es gibt auch verschiedene Bildungs- und Lehrprogramme: von Rechtschreibung und Mathematik bis zu Fremdsprachen. Dabei helfen farbige Bilder auf dem Bildschirm, den Stoff einleuchtender und interessanter zu gestalten.

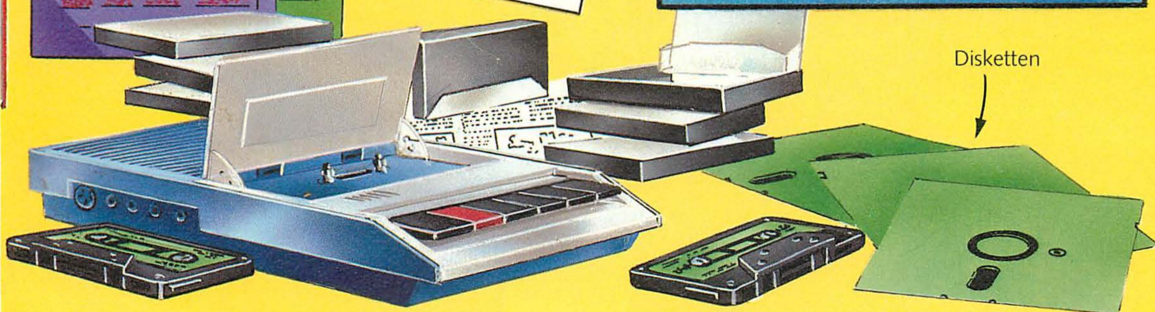
Hinweise

1. Es ist wichtig, daß das Programm in demjenigen BASIC-Dialekt geschrieben ist, den der Computer versteht.
2. Als Anhaltspunkt gilt, daß 1 KB Speicherplatz für ungefähr 40 Programmzeilen ausreicht.
3. Wenn ein Programm auf Band oder Diskette nicht läuft, obwohl es für den eigenen Computer geeignet sein sollte, dann sollte man das dem Händler sagen.

Klubs



Computer-Klubs bieten die Möglichkeit, mit anderen interessierten Leuten zu sprechen und Ideen und Programme auszutauschen. Mit Hilfe einer Computer-Zeitschrift läßt sich feststellen, ob es in der Nähe einen solchen Computer-Klub gibt.



Man kann Programme kaufen, die auf Kassette gespeichert sind. Dazu braucht man einen Kassettenrecorder. Es gibt auch Programme auf *Disketten* (Floppy disks). Diese Platten bestehen aus demselben Material wie Kassettenbänder, sind jedoch teurer, und man braucht ein Diskettenlaufwerk, um sie abzuspielen. Computer-Geschäfte verkaufen Kassetten und Disketten. Die Namen und Hersteller findet man in Zeitschriften.

Wie man seine eigenen Programme schreibt

Die meisten Mikrocomputer verstehen BASIC, eine vielseitig verwendbare Programmiersprache. Sie besteht aus Zeichen und Wörtern und ist leicht zu lernen. Am besten beginnt man damit, möglichst viele Programme zu lesen. Es ist sehr hilfreich, wenn man diese Programme dann auf einem Mikrocomputer ausprobieren kann. Gründlicher lernt man allerdings aus Büchern, die einen richtigen BASIC-Kurs bieten. Es gibt auch BASIC-Kurse auf Band oder Diskette, die man auf dem eigenen Mikrocomputer laufen lassen kann.

```
PROGRAMM LOSUNG

10 PRINT "BOSS, WIE HEISST DIE
   LOSUNG?"
20 INPUT L$
30 CLS
40 PRINT "HALT! WER DA? DIE
   LOSUNG!"
50 PRINT "DU KANNST ZWEIMAL
   RATEN"
60 FOR A=1 TO 2
70 INPUT A$
80 IF A$=L$ THEN A=2: GOTO 130
90 PRINT "FALSCH!"
100 NEXT A
110 PRINT "ZIEH LEINE, SPION!"
120 END
130 PRINT "TRITT EIN, MEIN FREUND!"
140 END
```

Wenn man das nebenstehende Programm durchliest, lernt man einige BASIC-Ausdrücke. Dieses Programm dient dazu, Spione zu entlarven...

Jede Programmzeile wird numeriert, am besten mit Zehnerzahlen. Diese Art der Numerierung bietet den Vorteil, daß man später zwischen zwei Befehlen leicht weitere Befehle einfügen kann. Der Computer arbeitet das Programm genau in der Zeilenfolge ab.

Um das Programm zu verwenden, wird es eingetippt. An jedem Zeilenende wird die RETURN-Taste (bei manchen Mikrocomputern die ENTER-Taste) gedrückt. Dann tippt man RUN, und der Computer arbeitet das Programm ab.

1 Ein Programm aufbauen

Sie testen das Go-Kart eines Freundes. Die Lenkung streikt, und das Fahrzeug rast geradewegs auf einen Teich zu. Die Bremse kann nur durch einen bestimmten Code-Buchstaben ausgelöst werden; der Freund hat aber vergessen, uns diesen Buchstaben zu nennen. Man darf fünfmal raten, bevor man ins Wasser fällt.

2

1. Drucke Titel und Anweisungen
2. Wähle Zufallsbuchstaben
3. Frage Spieler nach einem Buchstaben
4. Wenn Buchstabe korrekt, drucke Hinweis und beende Programm
5. Wenn noch Zeit vorhanden, gib Hinweis und gehe zurück nach 3.
6. Wenn nicht, drucke PLATSCH und beende Programm

Der erste Schritt beim Schreiben eines Programms besteht darin, sich eine genaue Übersicht über das Programm zu verschaffen. Die Übersicht unter Nr. 2 gehört zu einem Computer-Spiel.

Danach zerlegt man den Vorgang in kleine Einzelschritte und überlegt, was der Computer jedesmal tun muß. Die Einzelschritte müssen genau in der richtigen Reihenfolge aufgeschrieben werden.

PRINT bedeutet, der Computer soll alles auf den Bildschirm bringen, was zwischen Anführungszeichen steht.

INPUT bedeutet, der Computer erwartet eine Eingabe, die er an einer Stelle in seinem Speicher ablegen soll. Diese Stelle heißt C$.
$ bedeutet eine Reihe Buchstaben (ein Wort).

CLS bedeutet: Bildschirm räumen.

FOR...TO sagt dem Computer, wie oft er die Befehle in den Zeilen 60 bis 100 ausführen soll.

IF...THEN sagt dem Computer, was er tun soll, wenn (if) eine bestimmte Bedingung erfüllt ist. In diesem Fall sagt ihm
GOTO, er soll mit Zeile 130 fortfahren. Wenn die Bedingung nicht erfüllt ist, fährt der Computer mit Zeile 90 fort.
END sagt dem Computer, daß er alles erledigt hat und das Programm zu Ende ist.

```
HALT! WER DA? DIE LOSUNG!
DU KANNST ZWEIMAL RATEN
? KARTOFFELCHIPS
FALSCH!
MIKROCHIPS
TRITT EIN, MEIN FREUND!
```

Wenn das Programm läuft, fragt der Computer nach dem Codewort und speichert es. Dann fordert er dazu auf, das Codewort zu raten. Das Wort INPUT in Zeile 70 läßt ein Fragezeichen auf dem Bildschirm erscheinen; das bedeutet, der Computer wartet auf die Antwort. Er vergleicht sie mit dem Codewort in seinem Speicher. Wenn beide übereinstimmen, erscheint TRITT EIN, MEIN FREUND.

3

```
10 PRINT "SPIEL PLATSCH"
20 PRINT
30 PRINT "DIE LENKUNG DES GOKARTS"
40 PRINT "IST AUSGEFALLEN.SIE"
50 PRINT "FAHREN AUF EINEN TEICH"
60 PRINT "ZU.DIE BREMSEN GREIFEN,"
70 PRINT "WENN DER RICHTIGE BUCH-"
80 PRINT "STABE EINGEGEBEN WIRD."
90 PRINT "5 VERSUCHE SIND MOEGLICH."
100 LET C$=CHR$(64+INT(RND(1)*26+1))
110 FOR G=1 TO 5
120 INPUT G$
130 IF G$=C$ THEN G=5 GOTO 210
140 IF G$<C$ THEN PRINT "DANACH!"
150 IFG$>C$ THEN PRINT "DAVOR!"
160 PRINT G$
170 NEXT G
180 PRINT "PLATSCH,KLATSCH,NASS"
190 PRINT "ETWAS FEUCHT, NICHT WAHR?"
200 END
210 PRINT "QUIIIIIETSCH..."
220 PRINT "RECHTZEITIG GEBREMST!"
230 END
```

Diese Programme laufen wegen der verschiedenen BASIC-Dialekte nicht auf allen Mikrocomputern. Am wahrscheinlichsten ist, daß die Befehle CLS (Bildschirm räumen) und RND (eine beliebige Zahl auswählen) geändert werden müssen. Wenn das Programm auf dem eigenen Computer nicht läuft, dann sollte man im zugehörigen Handbuch nachschauen.

4

```
SPIEL PLATSCH

DIE LENKUNG DES GOKARTS
IST AUSGEFALLEN.SIE
FAHREN AUF EINEN TEICH
ZU.DIE BREMSEN GREIFEN
WENN DER RICHTIGE BUCH-
STABE EINGEGEBEN WIRD.
5 VERSUCHE SIND MOEGLICH
? T
DAVOR!
T
? S
QUIIIIIETSCH...
RECHTZEITIG GEBREMST!
```

Nun überträgt man jede einzelne Anweisung in BASIC und gibt die Befehle Zeile für Zeile ein. Nach jeder Zeile sollte man überprüfen, ob alles richtig getippt ist.

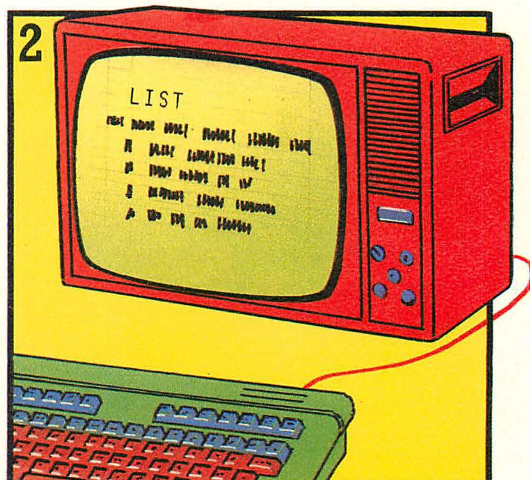
So sieht die Anzeige auf dem Bildschirm aus. Die Buchstaben hinter den Fragezeichen sind die Antworten.

Programme eingeben und laden

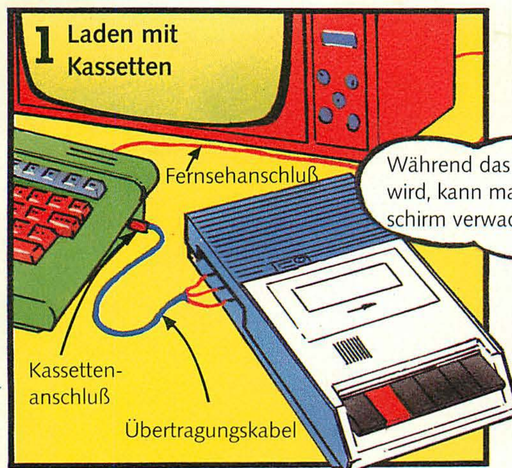
Alle Zeilen eines Programms, die man in den Computer eingibt, werden gespeichert. Anstatt das Programm einzutippen, kann man den Computer auch mit einem Programm „laden". Dazu verwendet man einen Kassettenrecorder. Hier sind einige Hinweise, wie man Programme eingibt und wie mit Kassetten geladen wird. Wenn das Programm nicht läuft, nachdem RUN getippt wurde, enthält es wahrscheinlich einen Fehler. Manchmal bewirken Fehler einen Zusammenbruch des Programms. Dann geht es nicht weiter, oder es geschehen unerwartete Dinge innerhalb eines Programms. Auf der gegenüberliegenden Seite sind die häufigsten Fehler zusammengestellt.



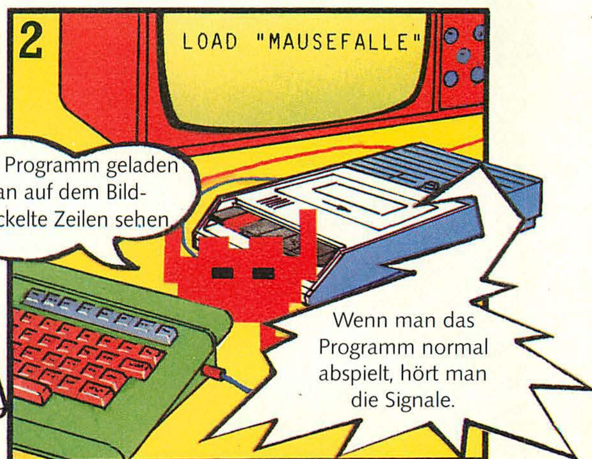
Richtige Zeichensetzung und Leerstellen sind genauso wichtig wie Buchstaben und Zahlen; deshalb muß ein Programm sehr sorgfältig eingegeben werden. Das Programm erscheint auf dem Bildschirm Zeile für Zeile, dadurch kann man es genau kontrollieren.



Startet man das Programm und es läuft nicht, dann gibt man LIST ein, damit es auf dem Bildschirm erscheint. Man kann es dann überprüfen, berichtigen und nochmals starten.



Ein Programm wird auf Kassettenband als Folge von hohen Pieptönen gespeichert. Um es in den Computer zu laden, schließt man den Kassettenrecorder, wie im Handbuch beschrieben, an den Mikrocomputer an. Lautstärke und Höhen müssen weit aufgedreht sein, damit der Computer die Signale aufnehmen kann.



Man tippt LOAD und in Anführungszeichen den Namen des Programms. Dann drückt man die Taste PLAY auf dem Kassettenrecorder. Jetzt wird das Programm in den Computer geladen. Dies kann einige Sekunden oder Minuten dauern, je nach Länge des Programms. Wenn das Programm nicht richtig geladen wird, kann das daran liegen, daß Lautstärke und Höhen nicht richtig eingestellt sind.

Fehler im Programm

Dieses Bild illustriert ein Programm mit vielen Fehlern. Die häufigsten Fehler sind Tippfehler. Wenn man BASIC nicht richtig eingibt, versteht der Computer die Befehle nicht. Diese Rechtschreibfehler nennt man *Syntax-Fehler* (SYNTAX ERROR).

Die meisten Mikrocomputer geben Fehlermeldungen aus, wenn etwas falsch eingegeben ist. Einige tun dies sofort, wenn der Fehler eingegeben wird, andere erst, wenn das Programm gestartet werden soll, also nach dem RUN- oder LIST-Befehl. Hier ein paar Beispiele:

MISSING ": Die Worte nach PRINT müssen in Anführungszeichen eingeschlossen werden.



SYNTAX ERROR: Es fehlt der PRINT-Befehl, der dem Computer sagt, daß er den Text auf den Bildschirm bringen soll.

CAN'T MATCH FOR: FOR...TO...NEXT sind Teile einer Anweisung, die dem Computer sagt, er soll diesen und die nächsten vier Schritte fünfmal ausführen. Das nennt man *Schleife*. Die NEXT-Anweisung, die die Zeilenzahl 80 haben sollte, hat keine Zeilenzahl; der Computer kann die Anweisung nicht verstehen.

SYNTAX ERROR: Das Wort ist falsch geschrieben, der Computer kann es nicht verstehen.

NO SUCH LINE: Es gibt keine Zeile 130 im Programm.

SYNTAX ERROR: Dies ist kein BASIC-Sprachelement, deshalb versteht es der Computer nicht. Der Befehl müßte heißen: END.

So sieht das korrekte Programm aus:

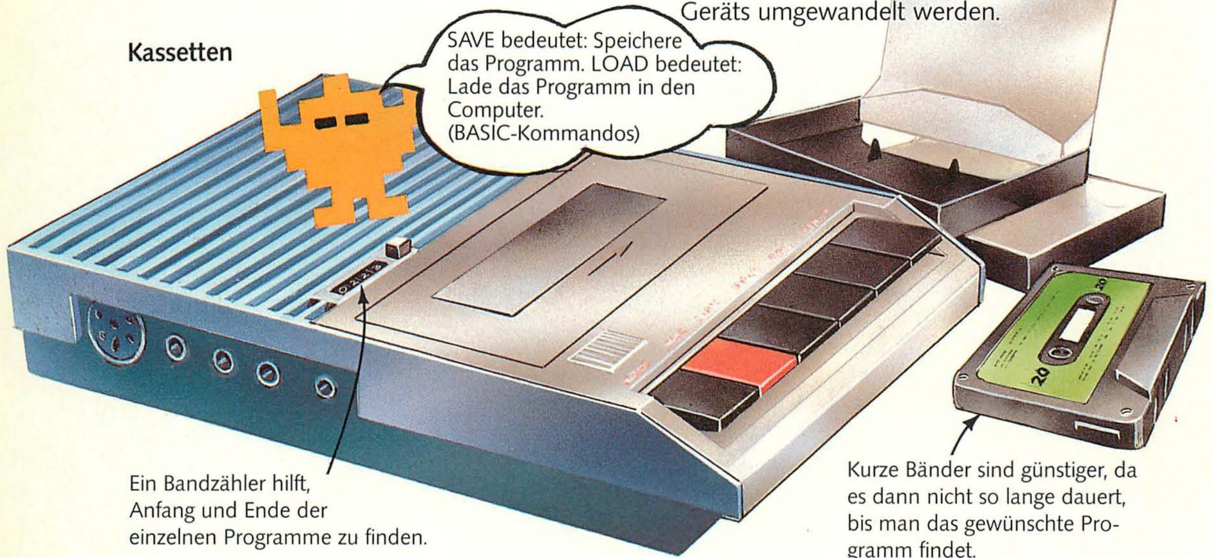
```
10 PRINT "WIEVIELE KROKODILE SIND IM
   FLUSS?"
20 PRINT "DU DARFST 5MAL RATEN."
30 LET A = 6
40 FOR N = 1 TO 5
50 INPUT R
60 IF R = A THEN N = 5: GOTO 110
70 PRINT "FALSCH"
80 NEXT N
90 PRINT "SCHNAPP! DU BIST AUFGEFRESSEN."
100 END
110 PRINT "RICHTIG. NUN PADDLE SCHNELL
    WEG!"
120 END
```

Wie man Programme abspeichert

Programme, die in einem Mikrocomputer gespeichert sind, können auf Kassettenband kopiert werden. Das ist sehr wichtig, denn beim Ausschalten des Computers würde sonst das Programm, das im RAM (Schreib-/Lesespeicher) gespeichert war, verlorengehen. Man kann Programme mit einem Diskettenlaufwerk auch auf Disketten speichern. Diese Methode ist vorteilhafter, wenn man viele Programme sammeln will. Man kann ein Programm mit einem Drucker auch auf Papier ausdrucken lassen.

Kassettenrecorder, Diskettenlaufwerke oder Drucker werden an einer bestimmten Buchse *(Port)* am Computer angeschlossen. Dazu gehört eine spezielle Schnittstelle *(Interface)*, in der die Maschinencode-Signale des Computers in die entsprechenden elektrischen Signale des benutzten Geräts umgewandelt werden.

Kassetten

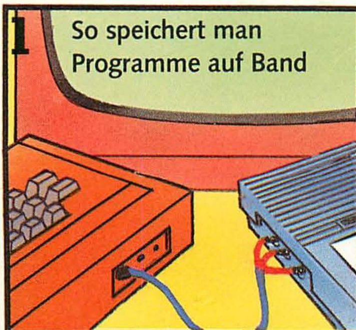


Ein Bandzähler hilft, Anfang und Ende der einzelnen Programme zu finden.

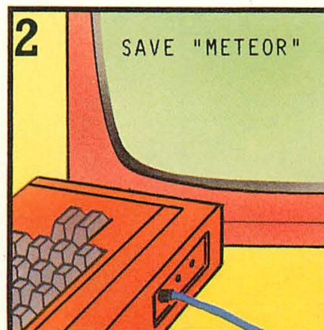
Kurze Bänder sind günstiger, da es dann nicht so lange dauert, bis man das gewünschte Programm findet.

Für die meisten Mikrocomputer läßt sich ein gewöhnlicher Kassettenrecorder verwenden; nur wenige Computer erfordern einen speziellen Recorder. Es gibt besondere *Datenbänder* für Programme, aber normalerweise genügt jedes andere Band von guter Qualität.

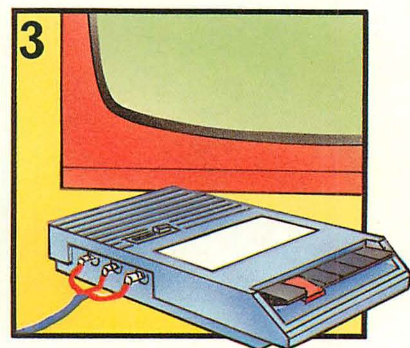
Programme auf Kassetten speichern und laden ist oft eine ziemlich verzwickte Sache. Wenn es nicht funktioniert, kann es daran liegen, daß der Tonkopf im Recorder gereinigt werden muß.



Mikrocomputer und Kassettenrecorder miteinander verbinden, wie im Handbuch vorgeschrieben. Dabei prüfen, ob die Kabel sich nicht kreuzen, sonst kann es zu Störungen kommen.

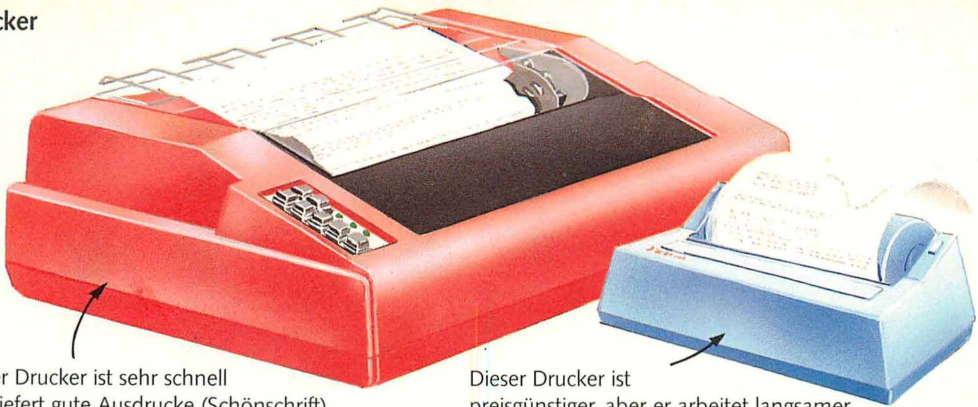


Danach SAVE und den Programmnamen in Anführungszeichen eingeben und die Tasten RECORD und PLAY auf dem Recorder drücken. Jetzt wird das Programm gespeichert.



Das Programm wird in Form magnetischer Impulse gespeichert, wenn das Band am Tonkopf des Kassettenrecorders vorbeiläuft.

Drucker



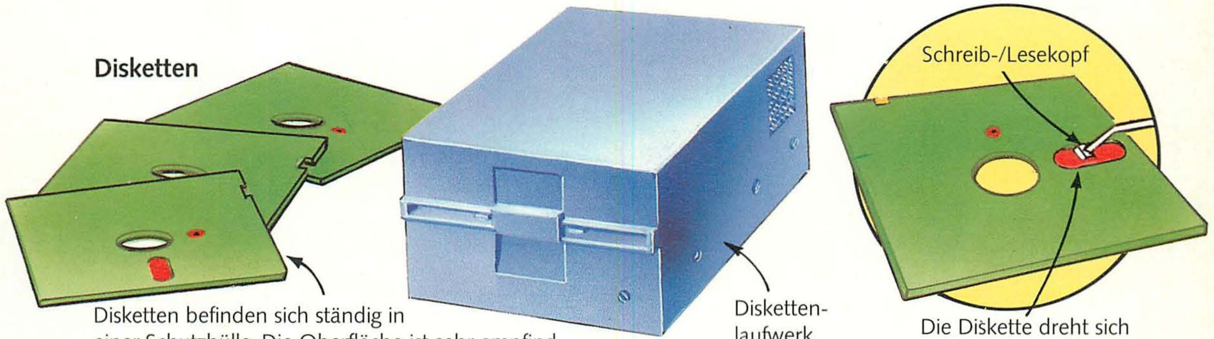
Dieser Drucker ist sehr schnell und liefert gute Ausdrucke (Schönschrift).

Dieser Drucker ist preisgünstiger, aber er arbeitet langsamer, und die Ausdrucke sind weniger gut lesbar.

Man kann Programme, Daten und manchmal auch Bilder ausdrucken, wenn man einen Drucker an den Computer anschließt. Die meisten Mikrocomputer haben als Anschluß eine standardisierte Schnittstelle (meist RS-232C oder V-24).

Informationen, die von einem Drucker gedruckt werden, heißen *hard copy*. Man kann von einem Programm viele Kopien machen, z. B. zur Verteilung an Freunde. Es gibt Drucker, die sehr schnell arbeiten. Manche Drucker drucken in einer Sekunde mehrere Zeilen; sie sind allerdings sehr teuer.

Disketten



Diskettenlaufwerk

Die Diskette dreht sich in der Schutzhülle.

Disketten befinden sich ständig in einer Schutzhülle. Die Oberfläche ist sehr empfindlich und würde bei Berührung beschädigt werden. Disketten speichern Programme ebenso wie Kassettenbänder. Die Oberfläche der Diskette ist glatt, ohne Rillen wie bei Schallplatten. Das Laden und Speichern von Programmen erfolgt im Diskettenlaufwerk, das man an den Mikrocomputer anschließt.

Die Diskette dreht sich im Diskettenlaufwerk. Dabei bewegt sich ein Schreib-/Lesekopf durch einen Spalt in der Schutzhülle sehr schnell über die Oberfläche der Diskette. Dieser Kopf kann alle auf der Diskette gespeicherten Daten lesen und auch neue Daten auf die Diskette schreiben.

Mehr über Speichern und Laden



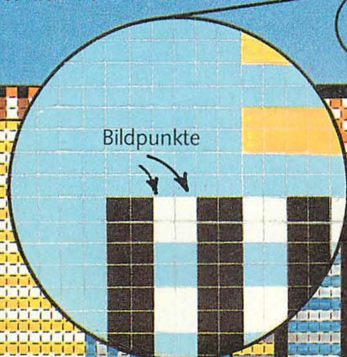
Wird ein Programm gespeichert, muß man ihm einen Namen geben und diesen in einem Verzeichnis notieren. Zum Laden muß dann der Name des Programms eingegeben werden. Wenn nur ein Buchstabe oder ein Zwischenraum falsch ist, kann der Computer den Namen des Programms nicht erkennen.

Wird der Mikrocomputer mit einem Programm vom Band oder von der Diskette geladen, so wird dieses im Speicher des Computers abgelegt. Man kann dann die Daten oder Programme beliebig verändern, ohne die auf Band oder Diskette gespeicherte Fassung zu verändern.

Bilder aus dem Computer

Auf dem Bildschirm werden Bilder aus vielen kleinen Bildpunkten (pixels) zusammengesetzt, indem der Mikrocomputer ganz kleine Felder einfärbt. Solche Bilder heißen Grafiken. Mit Hilfe entsprechender Programme kann man solche Grafiken anfertigen. Man kann auch – je nach Computertyp – mit einem Lichtstift direkt Bilder auf den Bildschirm malen oder das mit Hilfe eines speziellen Zubehörteils tun: mit einem Grafiktablett. Hier wird gezeigt, wie das funktioniert.

Bilder auf dem Bildschirm

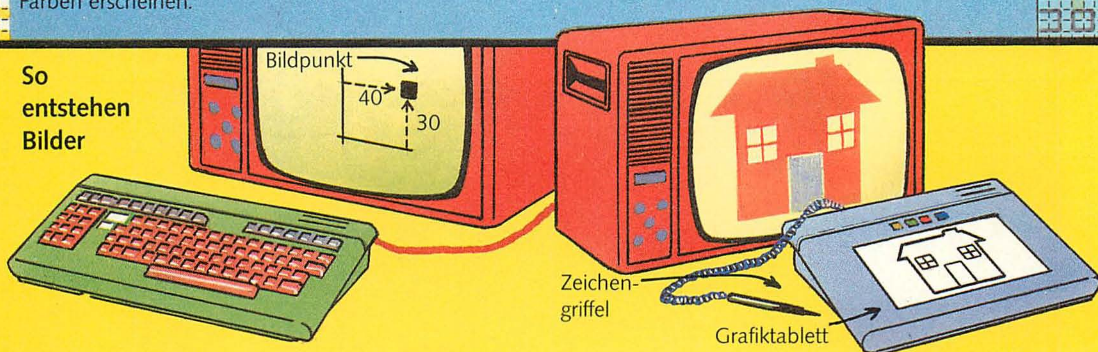




Betrachtet man Computerbilder aus der Nähe, dann kann man die Bildpunkte erkennen. Die meisten Computer können Farbbilder erzeugen, wenn man sie an ein Farbfernsehgerät oder einen Farbmonitor anschließt. Die Bilder entstehen dadurch, daß Gruppen von Bildpunkten in verschiedenen Farben erscheinen.

Auch Buchstaben, Zahlen und Symbole bestehen aus Bildpunkten. Der Mikrocomputer teilt den Bildschirm in Zeilen und Spalten auf. Jedes Zeichen entsteht dadurch, daß die Zusammenstellung bestimmter Bildpunkte aufgehellt wird.

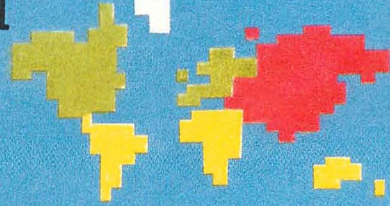
So entstehen Bilder



Zur Anfertigung eines Bildes macht man ein Grafik-Programm. Darin erhält der Computer Informationen darüber, welche Bildpunkte eingefärbt werden sollen. Dazu gibt man die Koordinaten des Bildpunkts an. Die Koordinaten zeigen, wie weit der Bildpunkt von einem Ausgangspunkt entfernt ist.

Ein Grafiktablett hat eine druckempfindliche Oberfläche, die mit einem Raster überzogen ist. Man legt die Zeichnung auf das Raster und zeichnet sie mit einem besonderen Zeichengriffel nach. Dadurch erhält der Computer automatisch die Koordinaten für alle Bildpunkte.

Die Bildqualität



Um jeden Bildpunkt einzeln anzusteuern, ist ein großer Speicher erforderlich. Deshalb fassen die meisten Mikrocomputer mehrere Bildpunkte zu einer Gruppe zusammen. Jede solche Bildpunktgruppe erhält eine getrennte Anweisung vom Mikrocomputer; alle Bildpunkte einer Gruppe haben die gleiche Farbe.



Je größer der Speicher eines Mikrocomputers ist, desto kleiner können die Bildpunktgruppen sein. Dadurch werden die Bilder realistischer. Solche Bilder nennt man *hochauflösende Grafik*.

Die Anzahl der Zeichen, die auf den Bildschirm passen, hängt von der Auflösung (Anzahl der möglichen Bildpunkte) ab; diese ist von Computer zu Computer verschieden. Bei einem Mikrocomputer, der den Bildschirm in 30 Spalten und 20 Zeilen unterteilt, kann man 30 Zeichen auf der Breite des Bildschirms und 20 Zeilen untereinander unterbringen.



Man kann auch direkt auf dem Bildschirm malen, wenn man ihn mit einem Lichtstift berührt. Beim Zeichnen einer Linie sendet der Lichtstift Signale an den Computer, damit dieser die Bildpunkte entlang der Linie aufhellt. Der Stift „sieht" den Strahl, der den Bildschirm erleuchtet, und „sagt" dem Computer die Positionen der Bildpunkte, die zu diesem Strahl gehören.



Der Mikrocomputer kann auch bewegte Bilder herstellen. Dies nennt man *Animation*. Dabei blinken die Bildpunkte in den verschiedenen Positionen des jeweiligen Gegenstandes auf dem Bildschirm. Dies geschieht so schnell, daß dabei der Eindruck von Bewegung entsteht.

Musik aus dem Computer

Die meisten Mikrocomputer können Melodien spielen und Geräusche machen, manche können sogar sprechen. Solche Computer haben einen speziellen Baustein (Chip), einen *Synthesizer*. Für manche Computer kann man den Synthesizer auch separat kaufen.

Töne werden erzeugt, wenn man Befehle wie SOUND oder BEEP, dann eine Zahl, die die gewünschte Note (z. B. C oder H) bezeichnet, und schließlich die gewünschte Dauer der Note eingibt.

1 Musik machen

Man kann Musikprogramme auf Kassetten kaufen. Einige Programme bringen zugleich mit der Musik Bilder auf den Bildschirm.

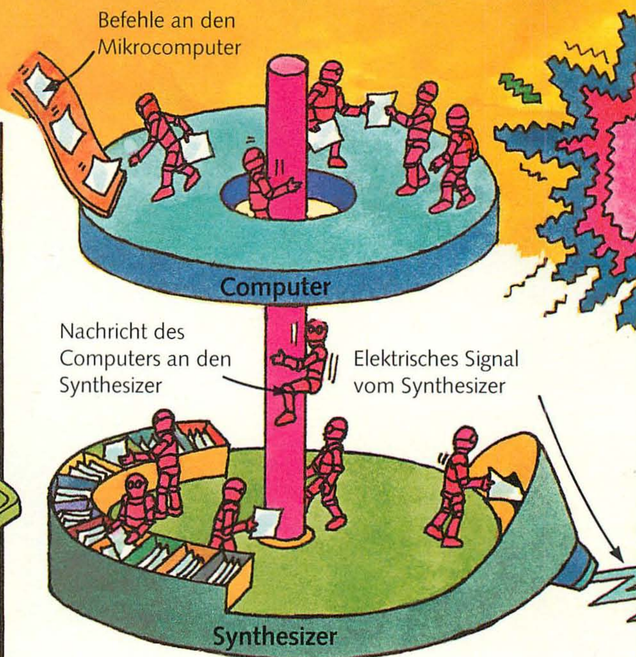


Man kann den Mikrocomputer so programmieren, daß er eine Melodie spielt. Dazu erhält er entsprechende Befehle für jede Note. Einige Mikrocomputer spielen sogar Akkorde und Harmonien. Diese Computer verfügen über mehrere „Stimmen", die so programmiert werden können, daß sie verschiedene Noten zur gleichen Zeit spielen.

2

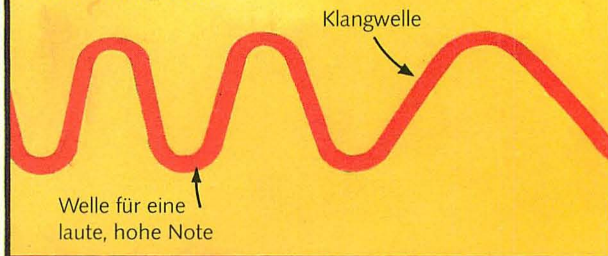


Man kann den Computer auch Noten lesen lassen, die man mit einem Lichtstift auf den Bildschirm zeichnet. Dazu gibt man dem Computer ein Programm ein, das Notenlinien auf den Bildschirm bringt. Auf diese Notenlinien kann man mit dem Lichtstift die gewünschten Noten setzen.

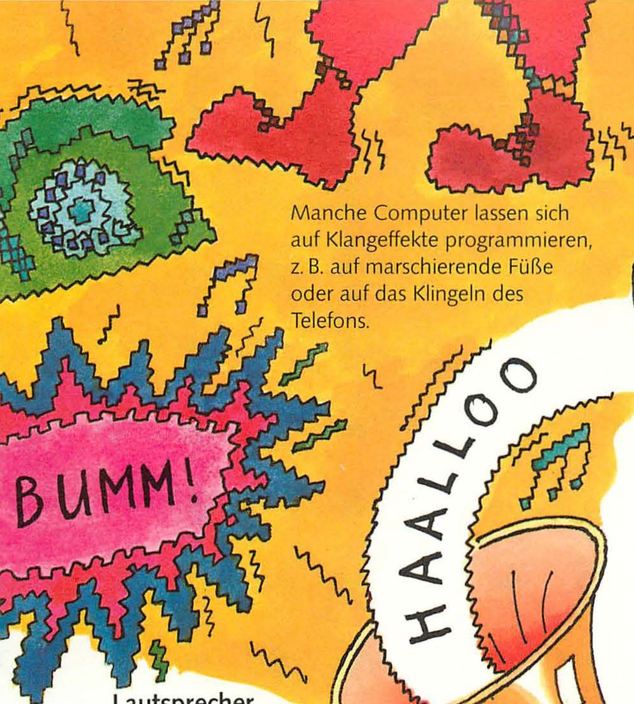


Wenn man dem Computer den Befehl gibt, Töne herzustellen, geschieht folgendes: Der Computer sendet im Maschinencode eine Nachricht an den Synthesizer, welcher Ton gewünscht wird. Der Synthesizer erzeugt ein elektrisches Signal, das in einem Verstärker ver-

Mehr über Töne

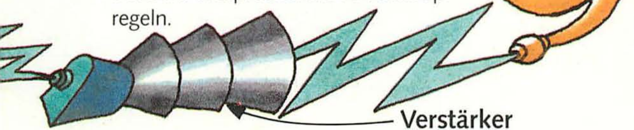


Die Schwingungen der Luft, die von einem Lautsprecher herrühren, heißen *Klangwellen*. Verschiedene Töne haben verschiedene Wellenformen. Eine laute, hohe Note z. B. hat große, eng zusammenliegende Wellen. Die Höhe der Wellen zeigt, wie laut die Note ist.

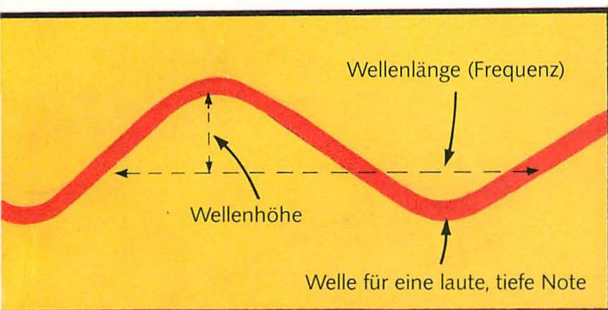


Manche Computer lassen sich auf Klangeffekte programmieren, z. B. auf marschierende Füße oder auf das Klingeln des Telefons.

Lautsprecher
Manche Mikrocomputer haben einen Lautsprecher in der Tastatur. Andere benutzen den Lautsprecher des Fernsehgeräts; in diesem Fall kann man die Lautstärke mit dem entsprechenden Drehknopf regeln.

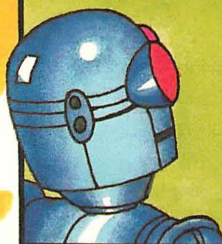
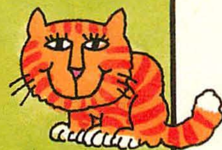


stärkt wird, und sendet dieses weiter zu einem Lautsprecher. Das Signal läßt den Lautsprecher vibrieren, dadurch entsteht der Ton. Unterschiedliche Signale des Synthesizers lassen den Lautsprecher verschieden schnell vibrieren: Dadurch entstehen verschiedene Töne.



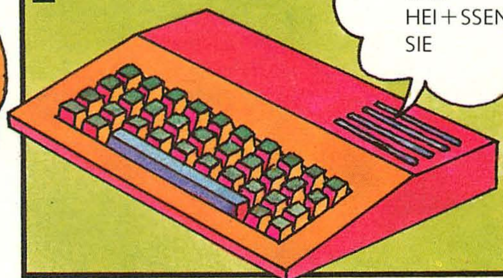
Die Länge der Wellen (von einem Wellenberg zum nächsten) gibt die Tonhöhe an. Die Veränderung von Lautstärke und Tonhöhe eines Klangs in bezug auf eine bestimmte Zeitspanne heißt *Klangform*.

1 Sprechende Computer

Einem Computer fällt es schwerer zu sprechen als Musik zu machen, denn das Aussprechen von Wörtern ist komplizierter. Die meisten Wörter bestehen aus verschiedenen Klängen, z. B. Ro - bo - ter. Mikrocomputer, die sprechen können, enthalten einen speziellen Chip, der die Laute der Wörter in Maschinencode gespeichert hat.

2



Der Computer formt Wörter, indem er mit Hilfe des Synthesizers Wortklänge zusammensetzt. Er folgt dabei grammatischen Regeln, die in seinem Speicher abgelegt worden sind. Das nennt man *Sprachsynthese*. Mikrocomputer mit Sprachsynthesizern sind nützlich für Blinde, die den Bildschirm nicht sehen können, oder für Kinder, die noch nicht lesen können.

3



Für Computer ist es sehr schwierig, Sprache zu verstehen. Sie müssen darauf programmiert sein, alle Wortklänge wiederzuerkennen. Da es verschiedene Stimmen und unterschiedliche Aussprachen gibt, können nur Computer mit riesigem Speicherplatz genügend entsprechende Informationen speichern.

Blick ins Gehäuse

Das Bild auf diesen beiden Seiten zeigt die Teile, die sich im Inneren eines kleinen Computers befinden. Alle Computer haben die gleichen Grundbausteine, die hier gezeigt werden. Allerdings sehen die meisten Computer etwas komplizierter aus und enthalten mehr Bauelemente.

Die wichtigsten Bauelemente des Computers sind die Chips; das sind hier die vier schwarzen Kästen „auf Beinen". Die gesamte Arbeit im Inneren des Computers wird von elektrischen Signalen geleistet, die durch die Chips pulsieren und durch die kupfernen Leiterbahnen fließen, die auf die Platine gedruckt sind. Mehr darüber, wie der Computer arbeitet, steht auf den nächsten Seiten.

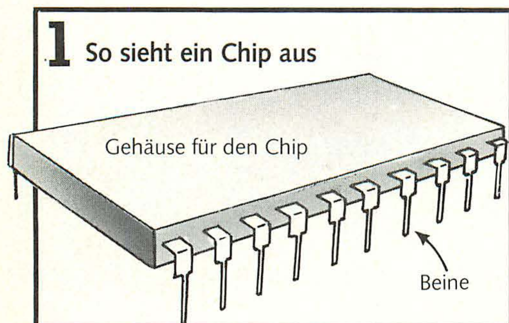
Das Netzteil
Das Netzteil liefert die Spannung, die der Computer zum Arbeiten braucht. Der Spannungsregler wandelt die Spannung der Stromquelle in die Betriebsspannung um.

Betriebs-ROM
Das Befehlsprogramm, das dem Computer sagt, wie er alles bearbeiten muß, ist hier dauerhaft gespeichert.

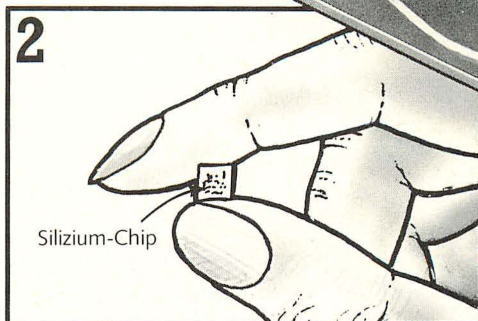
Gedruckte Schaltkreisplatinen
(engl. **P**rinted **C**ircuit **B**oard = PCB)
Auf der Oberfläche dieser Platine befinden sich stromleitende Stege, auf denen die elektrischen Signale von Chip zu Chip transportiert werden. Zu den Bauelementen gehören auch Kondensatoren und Widerstände, die den Stromfluß steuern helfen.

Widerstände

Betriebs-RAM
Dies ist der Schreib-/Lesespeicher, in dem Programme und Daten gespeichert werden.



Ein Chip ist ein kleines Gehäuse, das ein winziges Stück Silizium enthält. Die Oberfläche des Chips ist von weiteren Schaltkreisen durchzogen, die ganz winzig und sehr kompliziert sind. Die Metallbeine am Gehäuse des Chips leiten elektrische Signale zum Chip hin und von ihm weg.



Dieses Bild zeigt die tatsächliche Größe eines Chips. Er ist ungefähr so dick wie ein Fingernagel und kann bis zu zehn verschiedene Schaltkreise enthalten. Der richtige Name für Chip ist *integrierter Schaltkreis* (engl. **I**ntegrated **C**ircuit = IC).

Der Sinclair-Computer-Logik-Chip
Dies ist ein spezieller Chip, der besondere Anweisungen für diesen Computertyp enthält.

Anschlüsse für Fernsehgerät und Stromversorgung sowie für weitere Geräte, z. B. Kassettenrecorder oder Drucker.

Das Videoteil
Dieses Gerät wandelt die Signale des Computers in Videosignale um, die man mit einem Fernsehgerät empfangen kann.

Kondensator

Der Mikroprozessor
Dies ist die Zentraleinheit, das Steuer- und Rechenwerk des Computers. Sie führt die Befehle in den Programmen aus und steuert den Informationsfluß zum Schreib-/Lesespeicher und zum Bildschirm.

Steckverbindungen
Hier werden Zusatzgeräte angeschlossen, z. B. weitere Speicher oder Programmkassetten. Die elektrischen Signale zwischen Speicher oder Kassette werden über Metallstreifen am Rand der Platine weitergeleitet.

Die Kreise zeigen, wo die Schaltkreise durch die bedruckte Platine hindurchgehen und auf der anderen Seite fortgeführt werden.

Leistungsstärkere Computer

1

Leistungsstärkere Computer haben größere Speicher und mehr Chips. Im Bild sieht man die Platine eines Mikrocomputers mit ungefähr 40 Chips. Er hat mehrere ROM- und RAM-Chips, seine Speicherkapazität ist daher größer.

2

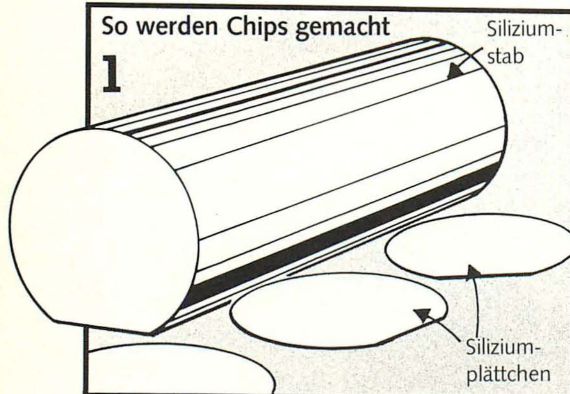
Ein leistungsstarker Computer, wie er in großen Firmen verwendet wird, hat Hunderte von Platinen, die mit Chips übersät sind. Die Platinen sind in Schränken untergebracht; diese Schränke können einen ganzen Raum füllen. Eine solche Anlage heißt *Großrechenanlage*. Sie kann viele verschiedene Aufgaben zur gleichen Zeit erledigen.

3

Ein Minicomputer ist eine kleine Rechenanlage mit weniger Platinen. Er ist meist auf bestimmte Arbeiten spezialisiert, z. B. auf Buchführung oder auf die Speicherung von Datenbankinformationen.

Blick in den Chip

Jeder Chip in einem Mikrocomputer hat Schaltkreise, die für bestimmte Aufgaben konstruiert sind. Das Bild rechts zeigt zwei Siliziumchips in starker Vergrößerung. Der eine ist ein Mikroprozessor, der andere ein ROM-Chip. Die Muster der verschiedenen Schaltkreise sind auf jedem Chip zu sehen. Die Schaltkreise sind so klein und kompliziert, daß bei der Herstellung von Chips bis zur Hälfte als Ausschuß ausgesondert werden muß.



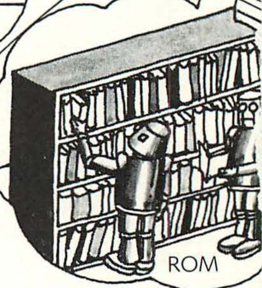
Chips werden aus hochreinen Siliziumkristallen hergestellt. Die stabförmigen Kristalle werden zu Scheiben zersägt, die einen Durchmesser von 100 mm haben und 0,5 mm dick sind. Jede Scheibe ergibt ungefähr 500 Chips. Silizium wird aus Quarzsand und Kohle hergestellt, daher sind Chips sehr billig.

Mikroprozessor

Ein Mikroprozessor wird manchmal „Computer auf einem Chip" genannt. Er hat verschiedene Arten von Schaltkreisen und kann wirklich die Arbeit eines kleinen Computers verrichten. Mikrocomputer werden häufig nach ihrem Mikroprozessor-Typ benannt.

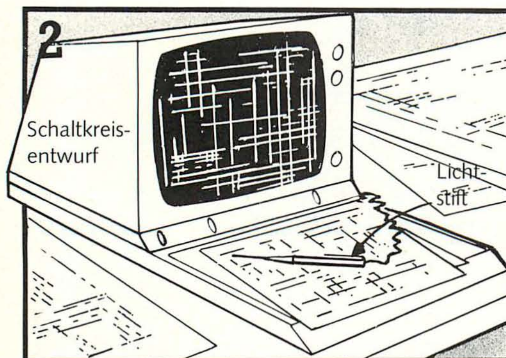
Dies sind RAM-Schaltkreise für die Zwischenspeicherung von Programmen. Hier wird die Information gespeichert, die der Mikroprozessor für eine bestimmte Aufgabe benötigt.



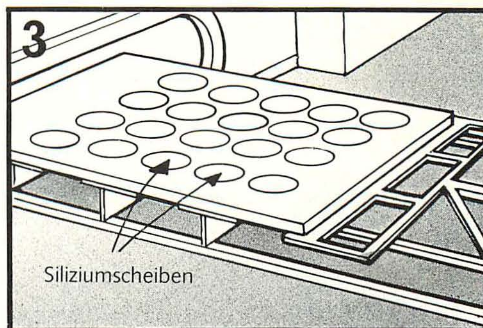


Dies ist ein ROM-Schaltkreis. Er enthält das Programm für die Funktionen des Mikroprozessors.

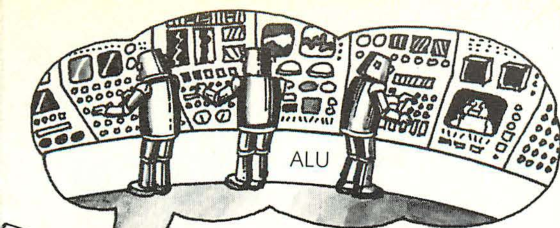
Die Schaltkreise auf den Mikroprozessoren sind durch sogenannte *Busse* verbunden. Busse stellen auch die Verbindung zwischen den Bausteinen auf der Platine und anderen Chips her.



Heute werden Schaltkreise für Chips bereits mit Hilfe von Computern entworfen. Hier wird ein Lichtstift dazu verwendet, um an einem Entwurf Änderungen vorzunehmen. Der nächste Schritt ist die Verkleinerung, damit der Schaltkreis auf einen Chip paßt.

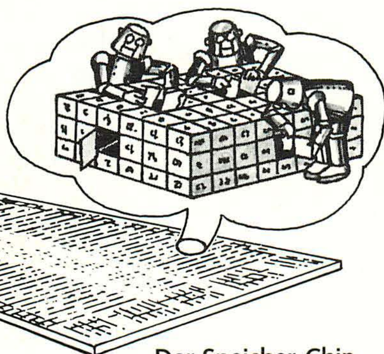


Die Schaltkreisentwürfe werden in einem fotografischen Prozeß auf die Chips aufgebracht. Die Siliziumscheiben werden in einen Brennofen gelegt. Dort werden die Schaltkreise auf chemischem Wege in das Silizium eingeätzt.



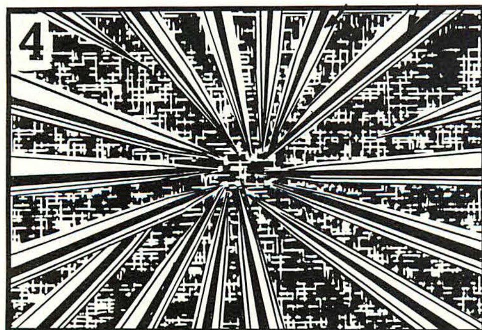
Die gesamte Rechenarbeit des Computers wird in den Schaltkreisen der Arithmetischen und Logischen Einheit (engl. **A**rithmetic and **L**ogic **U**nit = ALU) abgewickelt.

Die Uhr kontrolliert die Geschwindigkeit, mit der die Impulse durch den Mikroprozessor rasen.



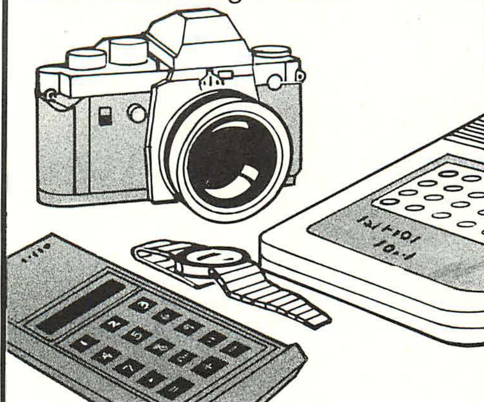
Der Speicher-Chip

Die Schaltkreise auf einem Speicher-Chip ähneln Hunderten von kleinen Kästchen. Auf einem ROM-Chip enthält jedes Kästchen ein wenig Information; auf einem RAM-Chip sind die Kästchen jedoch so lange leer, bis man sie mit Informationen füllt.



Auf einem Chip können viele verschiedene Schaltkreise eingeätzt werden. Dieser Prozeß kann mehrere Wochen dauern. Die fertigen Chips werden auf der Scheibe mit winzigen Sonden unter einem Mikroskop getestet. Fehlerhafte Chips werden markiert.

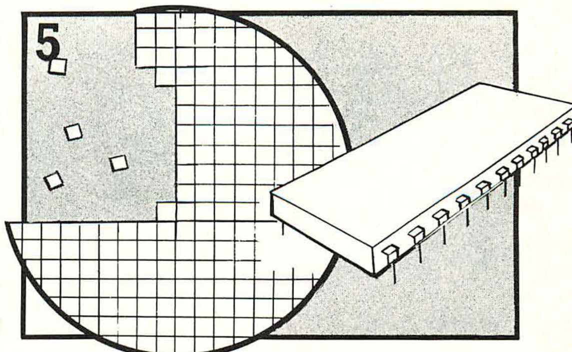
Weitere Einsatzmöglichkeiten



Mikroprozessoren werden in den verschiedensten Geräten als Kontrollmechanismus eingesetzt. Da sie sehr klein und leicht sind, kann man sie z. B. in Kameras, Uhren und Taschenrechnern unterbringen.



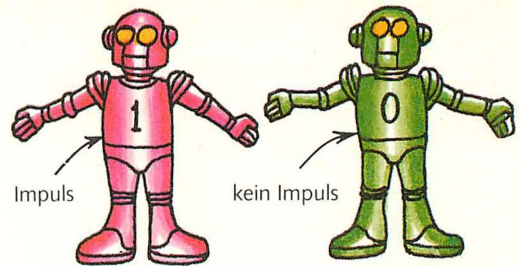
Mikroprozessoren haben bei vielen Geräten, etwa bei Waschmaschinen und Telefonvermittlungen, altmodische und sperrige elektromechanische Geräte ersetzt. Mikroprozessoren arbeiten wirtschaftlicher und sicherer.



Dann werden die Siliziumscheiben mit einer Diamantsäge in einzelne Chips zersägt. Die fehlerhaften Chips werden verworfen. Die unbeanstandeten Chips bekommen eine Schutzhülle, in der sie auf die Platine gesteckt werden können.

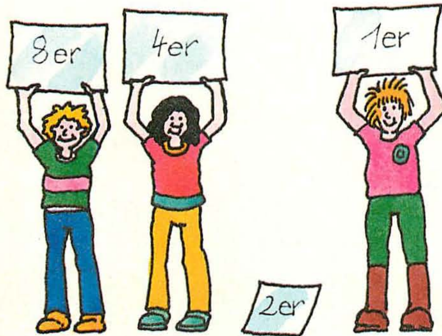
Wie Chips arbeiten

Ein Chip enthält viele Tausende elektronische Schaltkreise und Funktionselemente, durch die der Strom fließt. Gruppen von Transistoren bilden *Gatterschaltungen*. Das sind Schaltkreise mit einem oder mehreren Eingängen und einem Ausgang. Impulse an der Ausgangsleitung werden nur erzeugt, wenn an den Eingangsleitungen bestimmte Impulskombinationen vorliegen.



Der Maschinencode besteht nur aus zwei Signalen: „kein Impuls" oder 0 und „Impuls" oder 1. Das Binärsystem besteht also nur aus diesen beiden Ziffern. Jede Zahl wird durch eine Folge der Ziffern 1 und 0 ausgedrückt.

Zählen im Binärsystem



(**1**x8) + (**1**x4) + (**0**x2) + (**1**x1) = 13
13 entspricht der Binärzahl **1101**.

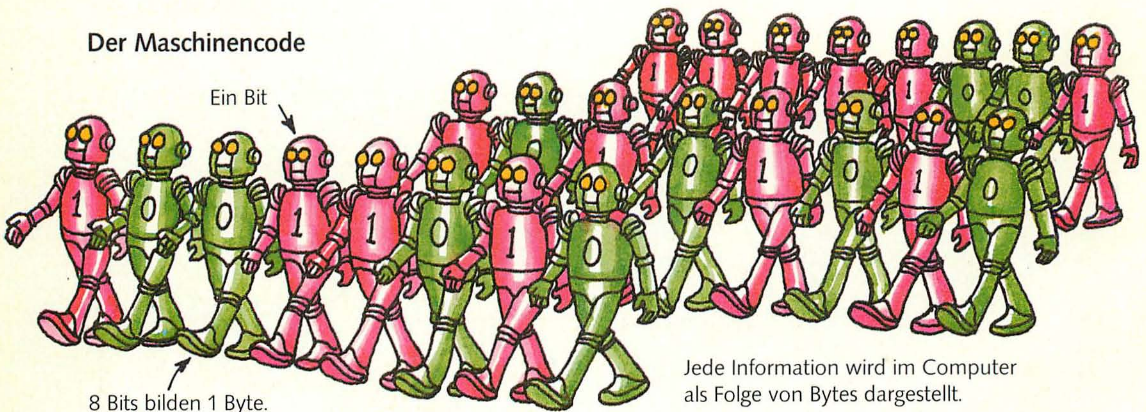


(**4**x1000) + (**0**x100) + (**2**x10) + (**1**x1) = **4021**

Binärzahlen bestehen aus den beiden Ziffern 0 und 1. Diese werden in Einer-, Zweier-, Vierer-, Achterspalten usw. geschrieben. Die Zahlen werden dadurch gebildet, daß Nullen und Einsen in die entsprechende Spalte geschrieben werden.

Das Dezimalsystem verwendet die zehn Ziffern 0 bis 9; es funktioniert nach dem gleichen Prinzip wie das Binärsystem: Die Dezimalzahlen werden in Spalten von Einern, Zehnern, Hunderten usw. geschrieben.

Der Maschinencode



8 Bits bilden 1 Byte.

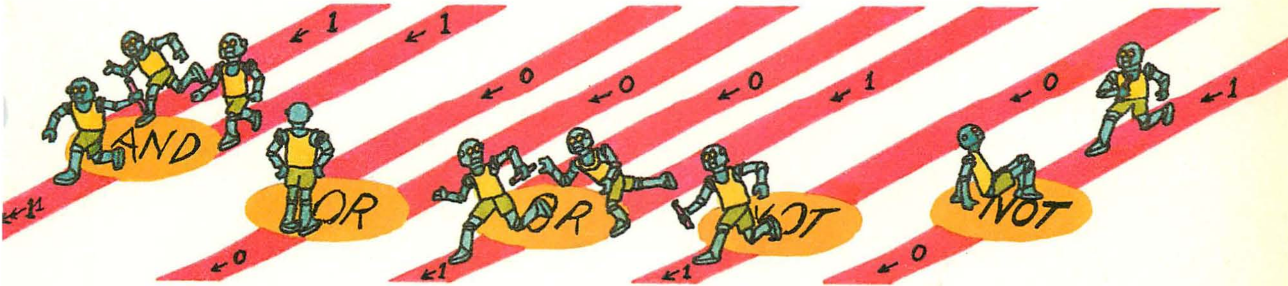
Jede Information wird im Computer als Folge von Bytes dargestellt.

Jedes einzelne Signal, 0 oder 1, heißt *Bit* (Abkürzung des engl. **b**inary dig**it**). Die meisten Heimcomputer verarbeiten die Informationen in Gruppen zu 8 Bits. Eine 8-Bit-Information nennt man 1 *Byte*.

8 Bits erlauben die Darstellung von 256 verschiedenen Codes oder Zeichen. Das ist ausreichend, um alle Symbole auf der Tastatur darzustellen; außerdem lassen sich noch einige weitere Informationen wie Farben oder Töne darstellen.

So verarbeitet der Computer die Informationen

Alle Informationen werden in binärer Form dargestellt und durch die Gatterschaltungen in binärer Form verarbeitet, geändert oder übertragen. Im folgenden werden drei grundlegende logische oder Gatterschaltungen vorgestellt. Sie werden mit anderen kombiniert, um weitere Funktionen darzustellen.



Der Ausgang eines AND-Gatters sendet nur dann „1" (Impuls), wenn die beiden Eingänge einen Impuls empfangen.

Der Ausgang eines OR-Gatters sendet „1", wenn einer oder beide Eingänge einen Impuls empfangen.

Der Ausgang eines NOT-Gatters sendet nur dann „1", wenn der Eingang keinen Impuls empfängt.

So addiert der Computer

Diese Bilder zeigen, wie der Computer eine bestimmte Kombination von Gatterschaltungen verwendet, um Binärziffern zu addieren (1+1, 1+0, 0+1, 0+0). Dies ist ein einfaches Beispiel; grundsätzlich verarbeitet der Computer jede Information durch solche Kombinationen von Gatterschaltungen.

So addiert der Computer 1+1.

AND-Gatter empfängt zwei Impulse und leitet einen weiter.

Jeder Impuls wird an beide Gatter weitergeleitet.

OR-Gatter empfängt zwei Impulse und leitet einen weiter.

AND

OR

NOT-Gatter empfängt einen Impuls und leitet keinen weiter.

NOT-Gatter empfängt einen Impuls, leitet ihn aber nicht weiter; der Impuls wird jedoch in der anderen Leiterbahn weitergeleitet.

NOT

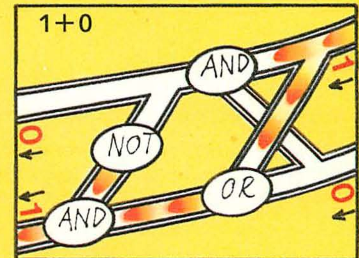
AND

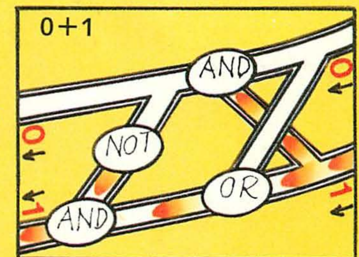
AND-Gatter empfängt einen Impuls, leitet aber keinen weiter.

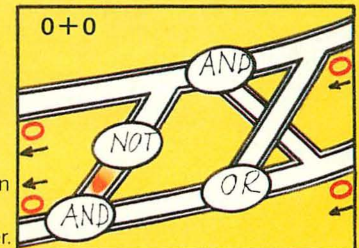
Leiterbahnen

1 0

Die Binärzahl 10 entspricht der Dezimalzahl 2.





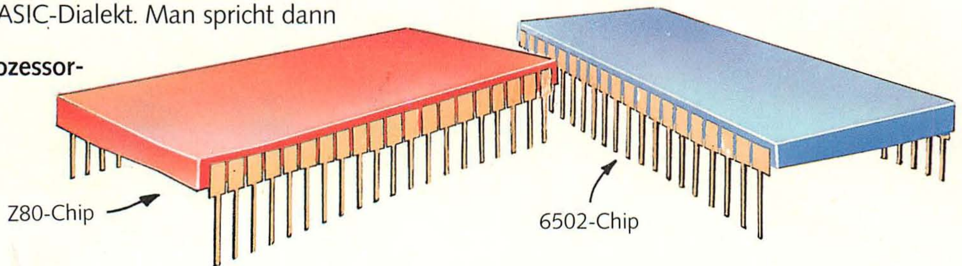


Hier wird dargestellt, wie andere Bits in derselben Gatterkombination addiert werden.

Mehr über Chips

Die Arbeitsweise eines Computers hängt von den verwendeten Chiptypen ab. Computer, die mit dem gleichen Mikroprozessor arbeiten, verstehen auch die gleiche Version des Maschinencodes. Der Interpreter, der BASIC in den Maschinencode übersetzt, ist im ROM gespeichert. Mikrocomputer, die den gleichen ROM-Chip haben, verstehen normalerweise denselben BASIC-Dialekt. Man spricht dann von der Kompatibilität der Software. BASIC entspricht mehr der menschlichen Sprache als der Maschinencode, deshalb braucht der Computer einen großen Interpreter. Sprachen wie BASIC heißen höhere Sprachen. Es gibt auch einfachere Sprachen, die mehr dem Maschinencode ähneln; diese sind für den Computer einfacher zu übersetzen.

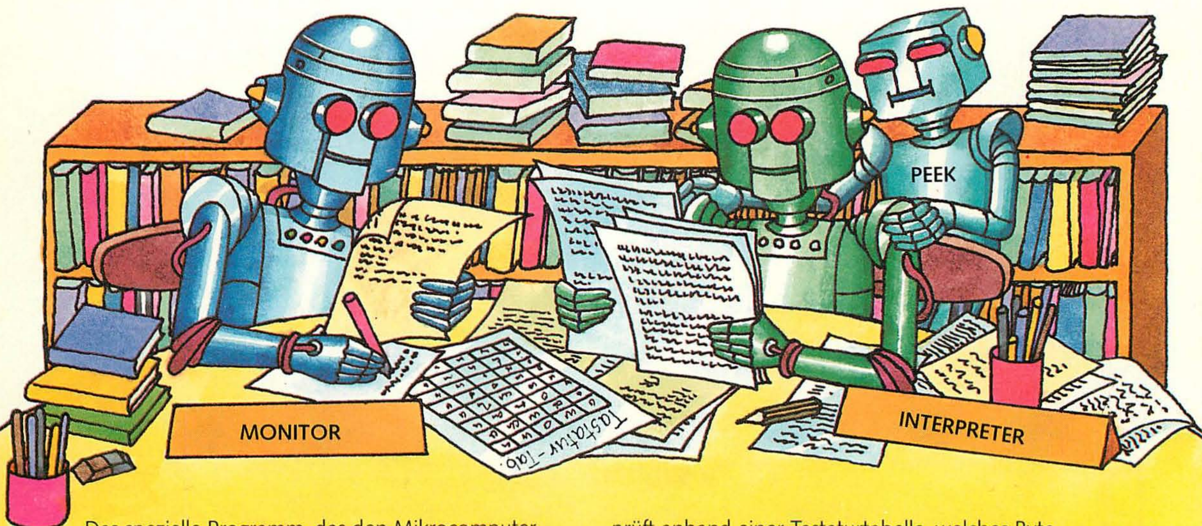
Mikroprozessor-Chips



Es gibt viele verschiedene Typen von Mikroprozessoren. Die in Heimcomputern meistverwendeten Mikroprozessoren sind die hier gezeigten Typen Z80 und 6502. Der Unterschied zwischen diesen beiden Mikroprozessoren liegt in ihrem inneren Aufbau; daher kann man sie von außen kaum unterscheiden. Die Rechenbefehle des Computers, die im ROM-Chip gespeichert sind, müssen in der richtigen Version des Maschinencodes geschrieben sein, z. B. im Z80-Maschinencode für den Z80-Mikroprozessor.

Was im Festwertspeicher geschieht

Der ROM-Speicher besteht aus winzigen Feldern mit Zahlenadressen; jedes Feld speichert 1 Byte Information. Der Mikrocomputer stellt die Bytes in einigen Feldern dar, wenn man PEEK und eine Adresse eingibt. Im Handbuch steht, für welche Adressen man PEEK eingeben kann. Das Byte erscheint dann auf dem Bildschirm als Dezimalzahl.



Das spezielle Programm, das den Mikrocomputer zum Laufen bringt, heißt *Monitor.* Es ist zusammen mit dem Interpreter im ROM gespeichert. Eine Aufgabe des Monitors ist es, herauszufinden, welche Taste auf der Tastatur gedrückt wurde. Er empfängt ein elektrisches Signal von der Taste und prüft anhand einer Tastaturtabelle, welches Byte diese Taste darstellt. Die meisten Mikrocomputer verwenden den gleichen Code. Er heißt ASCII. Das ist die Abkürzung für **A**merican **S**tandard **C**ode for **I**nformation **I**nterchange.

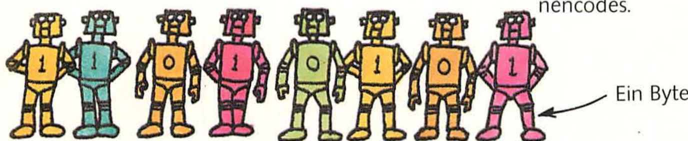
Was im Schreib-/Lesespeicher geschieht

System-Variable	Dieser Bereich enthält Informationen für den Mikrocomputer, z. B. wo das nächste Symbol auf dem Bildschirm erscheinen wird.
Programm-Bereich	Hier ist das Programm gespeichert.
Anzeige-Datei	In diese Datei kommt eine Maschinencode-Version der auf dem Bildschirm erscheinenden Daten und Ergebnisse.
Variable	Hier werden die Daten gespeichert.
Aktuelle Zeile und Arbeitsraum	Enthält die eingegebene Zeile
Rechner-Stapelspeicher	Hier erledigt der Mikroprozessor einige Berechnungen.
Freier Raum	Diesen Raum hat der Anwender zur Verfügung. Hier können auch andere RAM-Bereiche Platz „borgen", wenn sie voll sind.
Maschinen-Stapelspeicher	Der Mikroprozessor benutzt diesen Bereich als Speicherplatz, z. B. für die Zeilennummern der Programme.
GOSUB-Stapelspeicher	Hier ist die Programm-Zeilennummer gespeichert, zu der der Computer nach dem Unterprogramm zurückkehren muß.

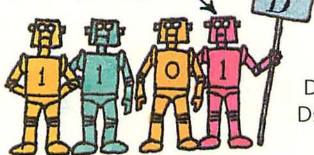
Der RAM-Speicher ist in verschiedene Bereiche aufgeteilt, die unterschiedliche Arten von Informationen speichern. Genau wie in ROM-Bereiche kann man auch für RAM-Bereiche PEEK eingeben. Wenn man die Bytes, die in einigen RAM-Bereichen gespeichert sind, ändern will, dann gibt man POKE, eine Adresse und eine Zahl ein. (Mit dem ROM-Speicher kann man das nicht machen, da der ROM ein Festspeicher ist.) Im Handbuch des Computers steht, für welche Bereiche des RAM man POKE eingeben kann, ohne daß dadurch die übrigen Aufgaben gestört werden. Normalerweise kann man mit POKE in einen Speicherplatz des Arbeitsspeichers laden; man kann damit auch Daten im Freiraum speichern, die dann mit PEEK abgerufen werden.

Einfaches Programmieren

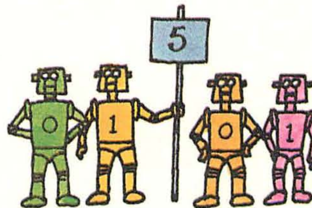
Ein im Maschinencode programmierter Computer kann die Befehle sofort ausführen, ohne sie vorher übersetzen zu müssen. Dies ist sehr nützlich, etwa bei schnellen Spieleprogrammen. Programmieren in Zweierzahlen ist allerdings kompliziert, deshalb benutzt man statt dessen andere einfache Codes wie Hex oder mnemonische Codes. Diese Codes ähneln schnellen Kurzschriftversionen des Maschinencodes.

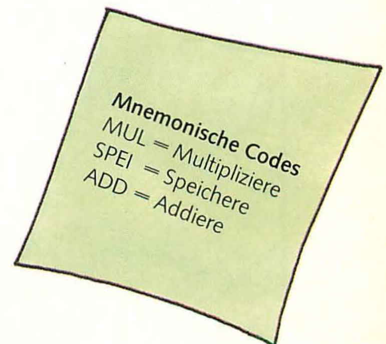


Der Dezimalwert der Gruppe ist 13 – in Hex D.



Das Byte D5 in Hex





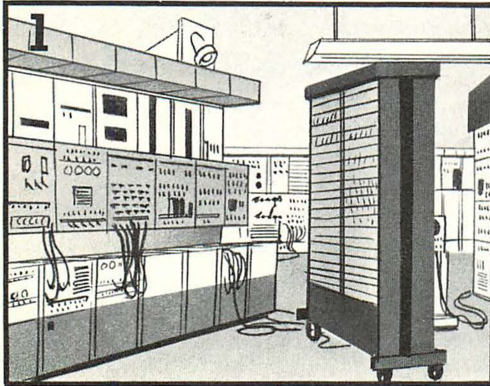
Hex (Abkürzung für Hexadezimal) ist ein Zahlensystem auf der Basis von 16. Ein acht-Bit Byte kann in zwei Hex-Ziffern geschrieben werden. Dazu teilt man das Byte in zwei Gruppen von je vier Binärziffern auf und verwandelt jede Gruppe in eine einzelne Hex-Ziffer. Die Dezimalzahlen 0 bis 15 werden dann durch die Ziffern 0 bis 9 und die Buchstaben A bis F dargestellt.

Ein mnemonischer Code ist eine Zusammenstellung von Abkürzungen, die für bestimmte Befehle an den Computer stehen. Jede Abkürzung löst eine bestimmte Reaktionskette im Computer aus. Einfache Codes können leichter in den Maschinencode übersetzt werden, daher braucht in diesem Fall der Interpreter nicht so groß zu sein.

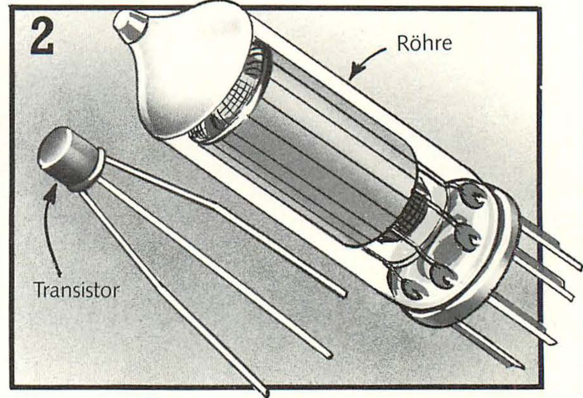
Von der Addiermaschine zum Mikrocomputer

Die ersten richtigen elektronischen Computer wurden im Zweiten Weltkrieg in England gebaut. Im Gegensatz zu früheren mechanischen Addiermaschinen waren sie programmierbar und hatten Speicher. Diese Computer wurden von Wissenschaftlern benutzt, um Codes zu knacken und Geschoßflugbahnen zu zeichnen. Viele Jahre lang wurden Informationen darüber streng geheimgehalten.

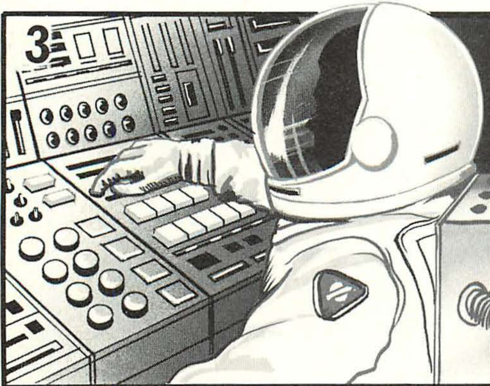
Nach dem Krieg begannen einige große Industrieunternehmen und einige Regierungen damit, Computer zu benutzen; sonst konnte sich niemand einen Computer leisten. Später wurden die Computer kleiner, billiger und immer leistungsfähiger. Dies führte zur Entwicklung der Mikrocomputer. Mikrocomputer können von jedermann benutzt werden, nicht nur von Wissenschaftlern.



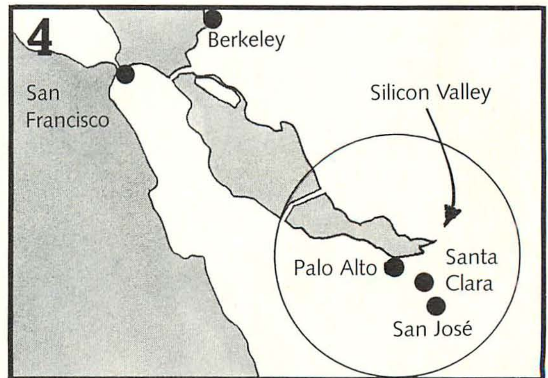
Die ersten Computer wurden gebaut, bevor Transistoren erfunden waren. Anstelle von Transistoren wurden damals Röhren verwendet. Diese waren ungefähr sieben Zentimeter lang und bestanden aus Glas. Für einen Computer brauchte man ungefähr 18 000 Röhren. Da sie häufig defekt waren, waren viele Ingenieure ständig damit beschäftigt, die beschädigten Röhren in den komplizierten Schaltkreisen zu finden und auszuwechseln.



Transistoren wurden Ende der 40er Jahre in den USA erfunden und in den 50er Jahren in zunehmendem Maße mit Erfolg eingesetzt. Sie leisteten genausoviel wie Röhren, waren aber erheblich raumsparender, billiger und arbeiteten schneller. Bald wurden in allen elektronischen Geräten die Röhren durch Transistoren ersetzt.



In den 60er Jahren mußte die amerikanische Regierung aufgrund des Wettlaufs in der Raumfahrt kleinere und leistungsfähigere Computer für ihre Raumschiffe entwickeln. Die Regierung finanzierte die Weiterentwicklung der integrierten Schaltkreise, die gerade erfunden worden waren. Sie bestanden aus mehreren Transistoren, die in einem winzigen Stück Silizium zusammengefaßt waren. Das nannte man Chip.



Silizium-Chips bedeuteten einen ernormen Durchbruch und führten zu einer neuen Wissenschaft, der Mikroelektronik. Das Forschungszentrum war Santa Clara Valley in Kalifornien, das später als Silicon Valley („Silizium-Tal") bekannt wurde. Die Mikroelektronik-Ingenieure sind bestrebt, immer mehr Bestandteile in einem Chip unterzubringen.

Computer-Generationen

ENIAC brauchte soviel Platz wie ein ganzes Haus und wog 30 Tonnen.

Die Baukosten betrugen über 500 000 Dollar.

Er brauchte 200 Kilowatt elektrische Leistung.

Er konnte zwei Zahlen in drei Millionstel Sekunden addieren.

Alle sieben oder acht Minuten war eine Röhre defekt.

1945

Die Geschichte des Computers hat sich bisher in vier Generationen abgespielt; jede Generation war kleiner und leistungsfähiger als die vorangegangene. Die erste Generation bildeten riesige Röhren-Computer. Einer davon war ENIAC. Er wurde nach zwei Jahren Bauzeit im Jahre 1945 fertiggestellt. Bei der zweiten Generation wurden Transistoren verwendet. Die dritte Generation verfügte bereits über Chips. Die Erfindung der Mikroprozessoren und die weitere Verkleinerung der Chips führten zur vierten Generation.

Ein Chip ist kleiner und dünner als eine Kontaktlinse.

Ein Chip kostet nur etwas mehr als 10 Mark.

Er kann zwei Zahlen in einer Zehnmillionstel Sekunde addieren.

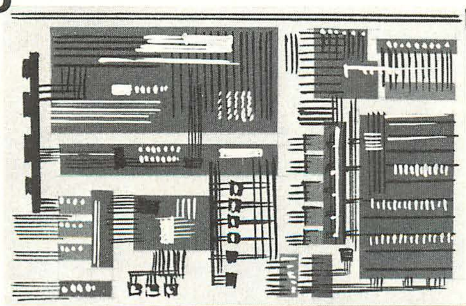
Er ist fast nie defekt.

Er braucht nur eine winzige Menge Strom.

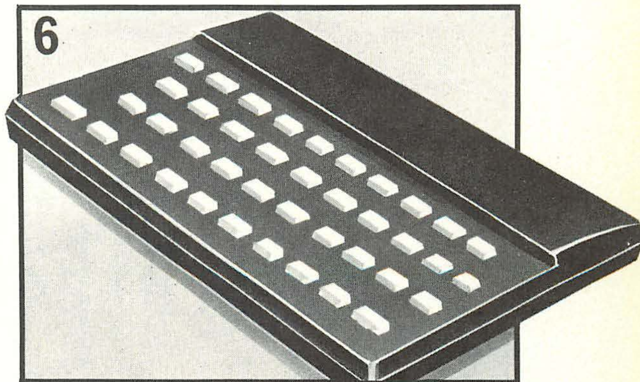


1980

5 Mikroprozessor-Chip



6



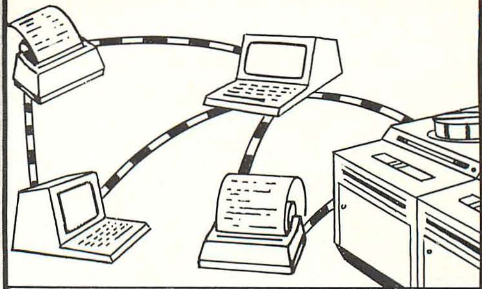
Einen bedeutenden Fortschritt gab es 1971, als es gelang, alle wesentlichen elektronischen Teile des Computers auf einem Chip unterzubringen. Dieser wurde Mikroprozessor genannt. Ein Computer-Schaltkreis, der früher mit Tausenden von Röhren ein ganzes Zimmer gefüllt hatte, konnte nun auf einem 5 mm großen quadratischen Silizium-Chip untergebracht werden.

Diese neue Technik führte zur Produktion von Mikrocomputern, die sich ein kleines Unternehmen oder sogar ein Privatmann leisten kann. Mikrocomputer kamen am Ende der 70er Jahre auf den Markt. Heute kann man Mikrocomputer kaufen, die so groß sind wie ein Taschenbuch und nur soviel kosten wie ein paar Röhren, wie sie in den ersten Computern eingebaut waren.

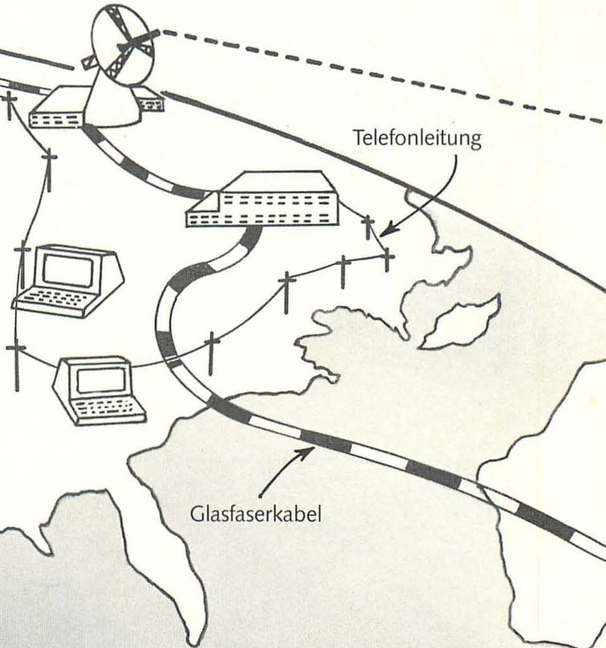
Computer-Netze

Wenn die notwendigen Verbindungen vorhanden sind und eine Methode zur klaren Signalübertragung besteht, kann man einen Mikrocomputer mit jedem anderen Computer in der Welt verbinden. Dabei kann man die bestehenden Kommunikationsmedien wie Telefon und Satelliten benutzen. Um sich zu „verstehen", brauchen Computer spezielle Programme; denn nicht alle Computer benutzen die gleiche Sprache oder den gleichen Dialekt, und sie arbeiten auch nicht alle mit der gleichen Geschwindigkeit. Die Verbindung von Computern dient dem Austausch von Informationen oder Programmen.

Netzwerke

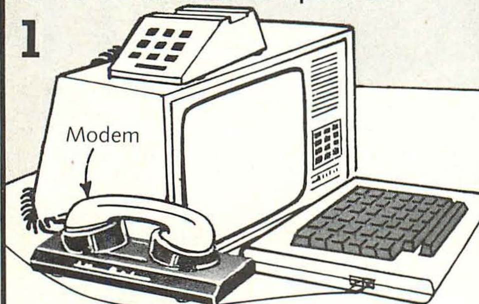


Computer können über Telefon zu *Netzen* oder *Netzwerken* verbunden werden. Auf dieser Seite sieht man, wie das geschieht. Man benötigt dazu ein Codewort, das den anderen Computer auf die Information vorbereitet. Man kann den Computer mit den verschiedensten Geräten verbinden: So können z. B. viele Computer denselben Drucker benutzen.



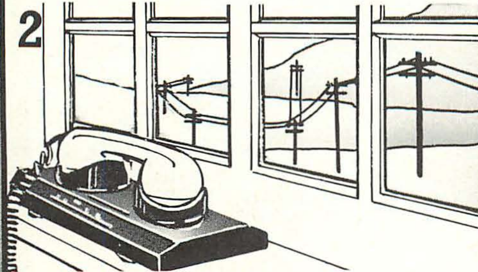
So wählt man einen Computer an

1



Man verbindet Mikrocomputer miteinander, indem man ein Telefon und ein sogenanntes *Modem* verwendet (Abkürzung für **Mo**dulator/**Dem**odulator). Es verwandelt Maschinencode-Signale in elektronische Signale, die durch ein Telefonkabel weitergeleitet werden.

2



Der Empfänger braucht ebenfalls ein Modem, das an seinen Mikrocomputer angeschlossen ist. Dieses Modem wandelt die ankommenden Signale wieder in den Maschinencode um.

Signale um die ganze Welt

Computer-Signale können in Form von Radiowellen mit Hilfe eines Satelliten rund um die Erde geschickt werden. Solche Satelliten übermitteln auch Ferngespräche und Fernsehprogramme in die ganze Welt.

Zur Zeit arbeitet man daran, elektrische Signale mit Lichtgeschwindigkeit durch Glasfaserkabel zu senden. Bei diesem System werden die Maschinencode-Signale in Lichtimpulse umgewandelt, die mit Hilfe von Glasfaserkabeln über Land oder unter Wasser an jeden beliebigen Punkt der Erde transportiert werden können.

```
*** FISCHHALLE ***

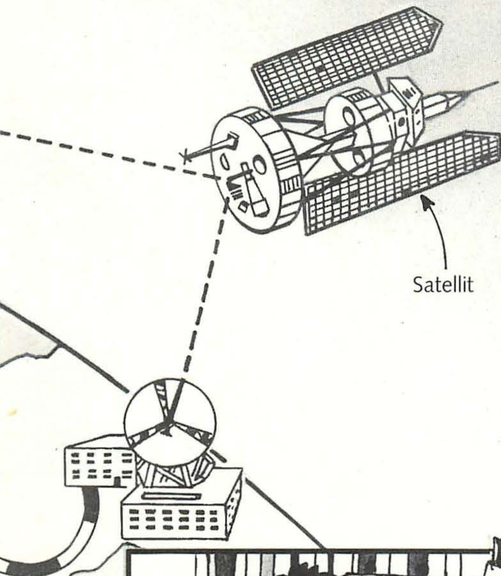
HEUTE FRISCH

1. KABELJAU      DM 13.90 /KG
2. KRABBEN       DM  8.90 /KG
3. MUSCHELN      DM  2.60 /KG
```

Mit Hilfe des Computers kann man auch einkaufen. Dazu verbindet man seinen Computer mit dem Computer des Geschäfts, in dem man einkaufen will. Man tippt ein, was man braucht, und gibt seine Kontonummer ein. Der Computer des Geschäfts sorgt dafür, daß man die gewünschten Waren bekommt, und informiert die Bank, damit das Konto entsprechend belastet wird.

```
LIEBER HANS,
     GEHST DU HEUTE ABEND
MIT INS KINO? NENA UND
UDO KOMMEN AUCH.
```

Mikrocomputer, die per Telefon miteinander verbunden sind, können als elektronische Briefkästen benutzt werden. Anstatt einen Brief zu schreiben und zur Post zu bringen, kann man ihn in den Computer eingeben, den Computer des Empfängers anwählen und den Brief auf dem Bildschirm des Empfängers hinterlassen. Dieses Verfahren ist sehr viel schneller als der übliche Postweg.





Geschäftsleute können zu Hause arbeiten, wenn sie einen Computer haben, der an das Netzwerk der Firma angeschlossen ist. Dadurch haben sie Zugang zu allen Dateien und können Mitteilungen an die Kollegen innerhalb des Netzwerks schicken.



Heutzutage werden immer mehr Mikrocomputer mit elektronischen Informationszentren zu Teletext-Systemen verbunden. Es gibt bereits ein weltweites Computer-Netzwerk, in dem Informationen aller Art gespeichert und ausgetauscht werden; damit steht einem jedes beliebige gespeicherte Wissen zur Verfügung.



In manchen Schulen gibt es für jeden Schüler einen Mikrocomputer, der mit einem zentralen Computer verbunden ist. Diesen benutzt der Lehrer, um zu überprüfen, was die Schüler tun, und um Programme einzugeben. Die Schüler können mit individueller Geschwindigkeit arbeiten und lernen.

Steuerung durch Mikrocomputer

Wenn die richtigen Anschlüsse vorhanden sind, können die meisten Mikrocomputer andere elektrische Geräte ebensoleicht steuern wie ihren eigenen Bildschirm oder Drucker. Dazu müssen die Maschinencode-Signale des Mikrocomputers in eine Form umgewandelt werden, die die angeschlossenen Geräte verarbeiten können. Diese Umwandlung erfolgt normalerweise am Steueranschluß. Die Steuerung selbst erfolgt durch den Mikroprozessor.

Steuerung einer Modelleisenbahn

Hier wird eine elektrische Eisenbahn von einem Mikrocomputer gesteuert, der durch ein Kabel vom Steueranschluß mit den Schienen verbunden ist. Er sendet Signale zu den Schienen und kann damit die Weichen stellen und den Zug stoppen oder anfahren lassen.

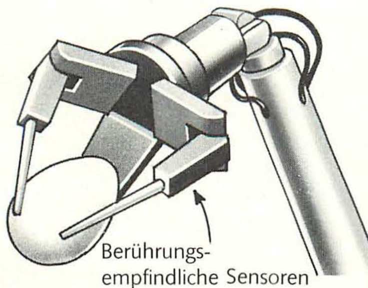


Der Mikrocomputer steuert die Geschwindigkeit des Zugs, indem er die Stromstärke variiert. Er kann auch die Runden zählen, die der Zug fährt, und er kann so programmiert werden, daß der Zug nach einer bestimmten Anzahl von Runden anhält.

Signale empfangen und absenden

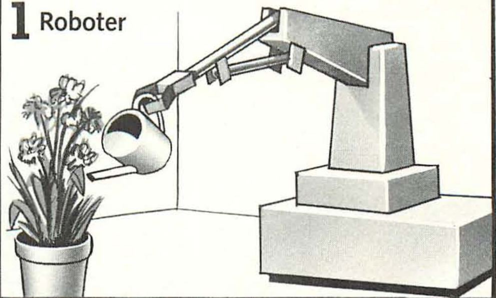
Stecke das Verbindungskabel in den Steueranschluß.

Der Mikrocomputer braucht eine Möglichkeit, Signale zu dem Gerät zu senden, das gesteuert werden soll, und eine „Erfolgsmeldung" zu empfangen. Der Steueranschluß enthält die Schnittstelle (Erklärung siehe Seite 16), die für diesen Informationsaustausch sorgt. Wenn der Mikrocomputer keinen Steueranschluß hat, kann man häufig einen als Zubehörteil kaufen.

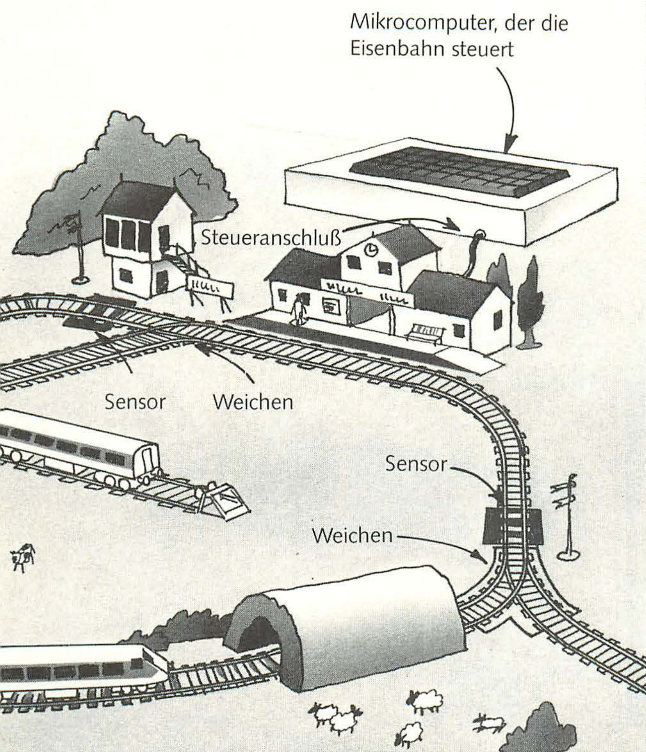


Der Mikrocomputer kann Sensoren benutzen, die ihm sagen, was gerade passiert, z. B. wenn er die Position eines Gegenstandes erfahren will, den er bewegen soll. Dazu kann ein Roboterarm mit berührungsempfindlichen Bereichen dienen, der vom Mikrocomputer gesteuert wird. Dieser macht eine „Mitteilung", sobald er mit etwas in Berührung kommt. Ein Roboterarm kann auch ein lichtempfindliches „Auge" haben.

1 Roboter



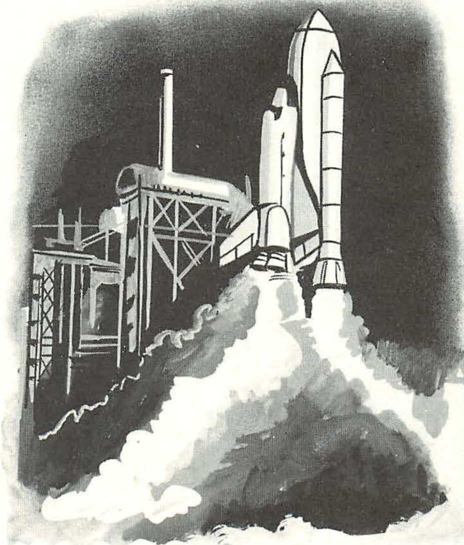
Manche Mikrocomputer kann man in Roboter verwandeln. Dazu verbindet man den Computer mit einem speziellen Arm und programmiert ihn so, daß er den Arm Bewegungen ausführen und Gegenstände aufheben läßt. Der Mikroprozessor im Computer handelt wie das „Gehirn" eines Roboters, indem er die Mitteilungen der Sensoren aus dem Arm für die nächstfolgende Bewegung auswertet.



Wenn der Zug einen druckempfindlichen Sensor überfährt, erhält der Computer die Mitteilung, an welcher Stelle der Gleise sich der Zug befindet und welcher Weiche er sich nähert. Der Computer hat ein Programm, das ihm sagt, was als nächstes zu tun ist.

Mikrocomputer im Weltraum

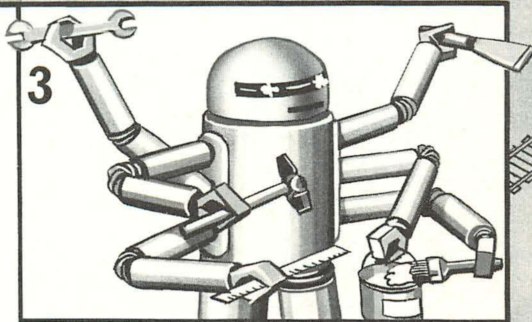
In der Raumfähre befindet sich ein Mikrocomputer, der mit einem normalen Heimcomputer vergleichbar ist. Unbemannte Raumschiffe wie „Viking" und „Voyager", die zum Mars und zum Saturn geflogen sind, wurden ebenfalls von Mikrocomputern gesteuert, die sich an Bord befanden und über Radiowellen mit großen Computern auf der Erde verbunden waren.



Diese Mikrocomputer führen komplizierte Rechnungen sehr schnell aus. Sie zeichnen Kurse, steuern den Antrieb und den Treibstoffverbrauch. Sie können Experimente machen und Fotos aufnehmen. Sie geben Berichte zur Erde und erhalten von der Erde neue Anweisungen.



In Fabriken werden größere Roboter eingesetzt. Sie erledigen viele Aufgaben – vom Bewegen schwerer Autokarosserien bis zum Zusammensetzen winziger mechanischer Teile. Es sind Roboter und nicht nur Maschinen: Man kann sie für verschiedene Arbeiten programmieren, und sie können eigene Entscheidungen treffen.



„Robot" ist das tschechische Wort für Schwerarbeit. Der tschechische Schriftsteller Karel Čapek hat es zuerst in den 20er Jahren zur Bezeichnung von künstlichen Menschen benutzt. Roboter können für eintönige oder gefährliche Arbeiten eingesetzt werden. Da sie nicht atmen, können sie im Weltraum arbeiten oder in Bergwerken, wo oft giftige Gase auftreten.

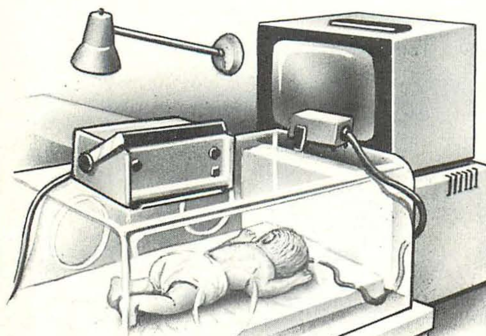
Weitere Einsatzgebiete

Mikrocomputer werden in den verschiedensten Arbeitsbereichen eingesetzt. Sie sind klein und leistungsfähig und können jede beliebige Information verarbeiten, sobald diese in Maschinencode umgewandelt vorliegt. Sie können weitaus schneller rechnen als der Mensch. Sie können sehr viele Informationen auf kleinstem Raum speichern, und sie haben im Gegensatz zum Menschen ein absolut lückenloses Gedächtnis.

Mikrocomputer in der Medizin

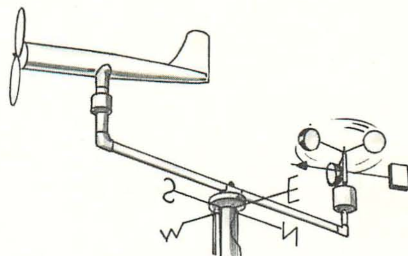
```
HABEN SIE HERZBESCHWERDEN?
JA.
SEHEN SIE SCHLECHT?
JA.
HATTEN SIE IN LETZTER ZEIT
MIGRAENE?
NEIN.
```

Mikrocomputer können Krankengeschichten speichern, und von manchen Ärzten werden sie zur Diagnose eingesetzt. Der Patient gibt die Antworten auf bestimmte Fragen ein, und der Computer vergleicht die Antworten mit Tabellen in seinem Speicher. Er gibt dann wahrscheinliche Diagnosen und Behandlungsvorschläge aus.



Die Mitarbeiter einer großen Klinik in Berlin haben ein Mikrocomputer-System entwickelt, das zu früh geborene Babys versorgt, die Schwierigkeiten beim Atmen haben und an eine künstliche Lungenmaschine angeschlossen werden müssen. Wenn zuviel Luft in die Lungen gerät, ist das sehr gefährlich, ebenso gefährlich wie zu wenig Luft, weil dann das Gehirn geschädigt wird. Der Mikrocomputer wacht darüber, daß die Lungen des Babys genau die richtige Menge Sauerstoff erhalten.

Wettervorhersage



Mikrocomputer in örtlichen Wetterstationen verarbeiten Daten, die sie von verschiedenen Meßgeräten erhalten, und senden die Ergebnisse an ein zentrales Wetteramt weiter.

Hilfe für Behinderte



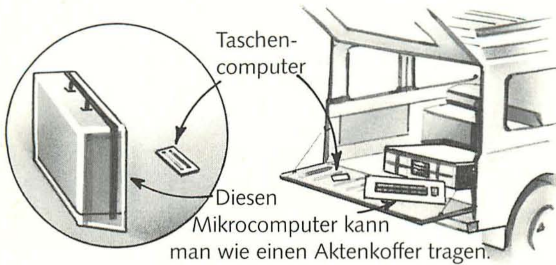
Taube oder stumme Menschen können Mikrocomputer zur Verständigung benutzen. Es gibt spezielle Tastaturen für teilweise gelähmte Menschen. Diese Tastaturen braucht man nur ganz leicht anzutippen, um einen Buchstaben oder ein Wort einzugeben.

Computer-Zeichnungen



Ein Mikrocomputer kann Gegenstände räumlich erscheinen lassen und diese Gegenstände so drehen, daß sie der Zeichner aus verschiedenen Perspektiven betrachten kann. Ein Architekt, der eine Brücke baut oder ein Gebäude entwirft, kann vom Mikrocomputer z. B. die Tragfähigkeit ausrechnen und ihn dann entscheiden lassen, ob das Bauwerk sicher ist. (Engl. **C**omputer **A**ided **D**esign = CAD)

Tragbare Mikrocomputer

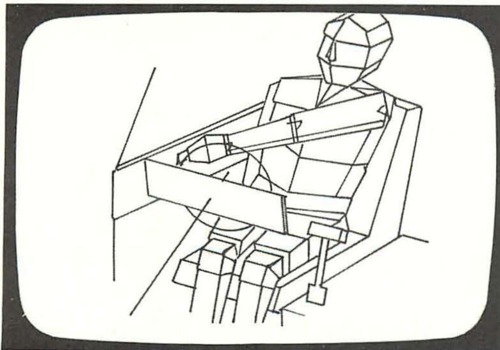


Menschen, die im Freien arbeiten, wie etwa Geologen, die nach Erdöl suchen, oder Vermessungsfachleute, benutzen einen tragbaren Mikrocomputer. Er kann an Ort und Stelle Daten speichern und verarbeiten.

In der Brauerei

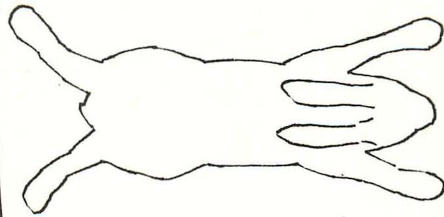


Auch in Brauereien und anderen Fabriken werden Mikrocomputer eingesetzt. Die Herstellung von Bier erfordert die richtige Mischung und Gärung bei genau vorgeschriebenen Temperaturen innerhalb bestimmter Zeiträume. Durch Sensoren erfährt der Computer, wann ein bestimmtes Stadium erreicht ist und ein anderer Vorgang beginnen kann.



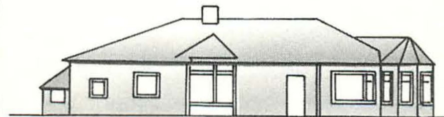
Ein so kleiner Raum wie das Innere eines Autos muß so gestaltet sein, daß der Fahrer alle Instrumente gut sehen und bedienen kann und genügend Platz hat. Zum Entwurf des Innenraums für ein neues Auto kann der Zeichner ein spezielles Programm einsetzen, das den Fahrer auf den Bildschirm zeichnet. Der Fahrer kann dann auf dem Bildschirm hin- und herbewegt werden.

Lernen mit Mikrocomputern

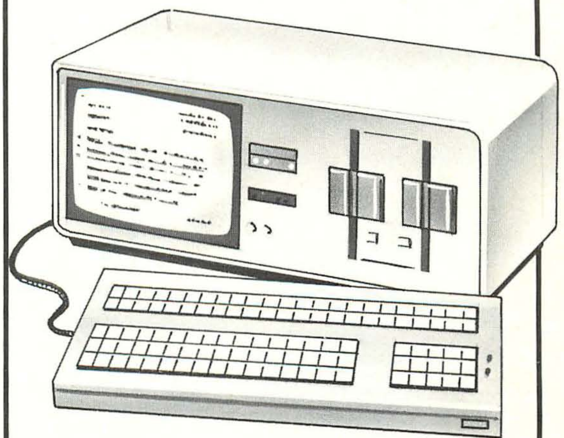


Mikrocomputer kann man auch im Unterricht einsetzen, z. B. um Französisch oder Navigation zu lehren. Man kann sogar ein Kaninchen auf dem Bildschirm „sezieren", ohne das Tier tatsächlich aufschneiden zu müssen.

Im Geschäftsleben



Selbständige Geschäftsleute und kleinere Unternehmen können einen Mikrocomputer für ihre Buchhaltung einsetzen. Ein Architekt oder Zeichner kann gut mit Mikrocomputer-Grafiken arbeiten.



Spezielle Textverarbeitungsprogramme werden in Büros verwendet, um Schreibarbeiten und Papier einzusparen. Mit einem Textprogramm kann man Standardbriefe und andere Texte eingeben, korrigieren und dann auf Diskette speichern, um sie bei Bedarf ausdrucken zu lassen.

Zubehör für den Mikrocomputer

Sobald man mit dem Mikrocomputer vertraut ist und weiß, was er leistet, kann man vielerlei Zubehör dazukaufen. Man nennt solche zusätzlichen Geräte, wie Diskettenlaufwerke, Drucker und Grafiktabletts, *Peripheriegeräte*. Um ein solches Gerät an den Mikrocomputer anzuschließen, braucht man eine Schnittstelle (Erklärung siehe Seite 16), die die Signale zwischen beiden Geräten entsprechend umwandelt. Jedes Gerät braucht seine eigene Schnittstelle. Die Schnittstellen für Kassettenrecorder und Fernsehgerät sind normalerweise bereits im Computer eingebaut. Viele Computer haben auch noch Schnittstellen für Drucker, Diskettenlaufwerk oder Lichtstift. Wenn nicht, kann man sie hinzukaufen. Viele Peripheriegeräte, insbesondere Drucker, Plotter (Zeichengerät) und Modems (für die Telefonverbindung mit anderen Computern), verwenden eine Standard-Schnittstelle, genannt RS-232C oder V-24.

Wenn man mehrere Peripheriegeräte an den Computer anschließen will, kann man eine spezielle Vorrichtung kaufen, in die man Platinen einstecken kann, die Schnittstellen für verschiedene Geräte enthalten.

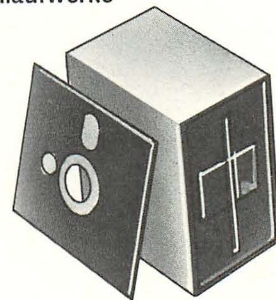
Zusätzliche Speicher



Manche RAM-Speicher können auf 64 KB erweitert werden, das heißt, man kann dann über 2 500 Programmbefehle speichern.

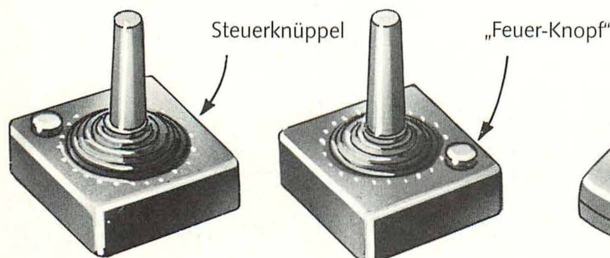
Bevor man sich weitere Peripheriegeräte anschafft, möchte man vielleicht eher den RAM-Speicher des Computers vergrößern. Dann kann man größere Programme für interessantere Spiele und bessere Grafik benutzen. Für manche Computer kann man zusätzlichen RAM-Speicher kaufen. Das sind Kassetten, die RAM-Chips enthalten und die man an den Computer ansteckt. Es gibt auch Computer, bei denen der Händler den zusätzlichen RAM-Speicher gleich an dem dafür vorgesehenen Platz auf der Platine einbaut.

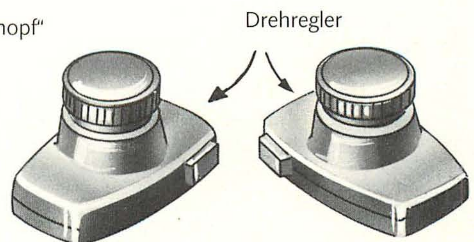
Diskettenlaufwerke



Zum Speichern von langen Programmen oder Datenbanken ist ein Diskettenlaufwerk besser als ein Kassettenrecorder, weil man die Daten dann schneller findet. Allerdings sind Diskettenlaufwerke auch erheblich teurer. Diskettenlaufwerke für Heimcomputer sind normalerweise für Minidisketten gemacht, die einen Durchmesser von 13,5 cm haben.

Steuerknüppel und Drehregler

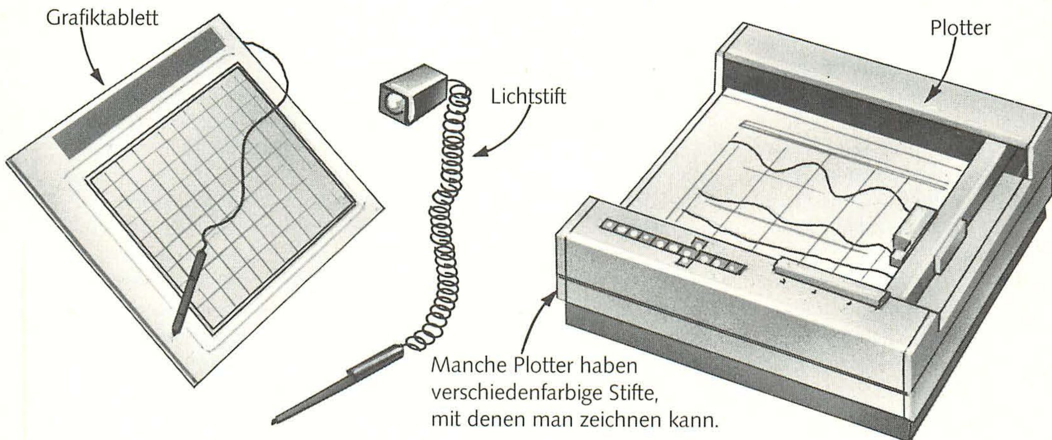




Steuerknüppel (Joysticks) und Drehregler braucht man für Computerspiele, wenn man Gegenstände wie Raumschiffe oder Flugzeuge auf dem Bildschirm bewegen will. Man kann zu diesem Zweck auch bestimmte Tasten auf der Tastatur verwenden, aber mit Steuerknüppeln und Drehreglern kann man besser steuern, und es macht mehr Spaß. Mit einem Steuerknüppel kann man den Gegenstand in jede Richtung bewegen, Drehregler bewegen ihn dagegen nur nach rechts, links, aufwärts oder abwärts. Steuerknüppel haben meistens einen „Feuer-Knopf“, mit dem man Flugkörper abschießen kann. Die „Maus“ ist eine präzisere Version des Steuerknüppels.

Grafik

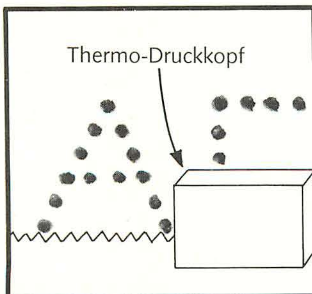
Wer sich für Grafik interessiert, kann gute Bilder auf den Bildschirm bringen, wenn er diese auf ein Grafiktablett zeichnet. Man kann auch gute gedruckte Bilder erhalten, wenn man einen Plotter einsetzt. Dabei wird ein Schreibstift über ein Stück Papier geführt; die Bewegungen des Stifts werden vom Computerprogramm gesteuert. Dies sind allerdings teure Geräte, daher fängt man besser mit einem Lichtstift an, der viel billiger ist.



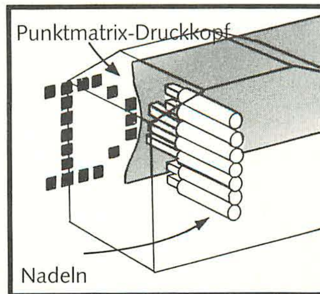
Für einige Computer kann man eine hochauflösende Grafikkarte oder Kassette kaufen, um die Bildqualität zu verbessern. Damit stehen einem mehr Farben zur Verfügung, die einzelnen Farbpunkte werden kleiner, und die Zeichnung wird genauer. Auch die Buchstaben und Ziffern können dann kleiner werden, und man bringt mehr Text auf dem Bildschirm unter. Da die hochauflösende Grafik sehr viel Speicherplatz in Anspruch nimmt, braucht man auch mehr RAM-Speicherplatz.

Mehr über Drucker

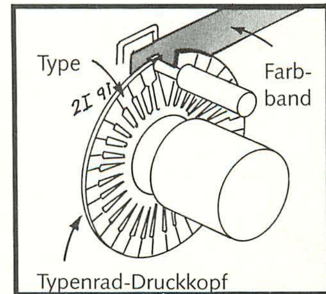
Es gibt drei Arten von Druckern: *Thermo-*, *Punktmatrix-* und *Typenraddrucker*. Thermodrucker sind am billigsten, und obwohl der Druck ziemlich unsauber sein kann, genügen sie zum Ausdruck von Programmen.
Punktmatrixdrucker sind ebenfalls recht preiswert. Typenraddrucker sind teuer, dafür ist der Druck von sehr guter Qualität. Bidirektionale Drucker drucken eine Zeile auf dem „Hinweg" und eine weitere Zeile auf dem „Rückweg". Dadurch wird Zeit gespart.



Thermodrucker erzeugen heiße Punkte und berühren damit hitzeempfindliches Papier. Es verfärbt sich dort, wo es erwärmt wird. Buchstaben werden durch Muster aus dunklen Punkten gebildet.



Punktmatrixdrucker haben einen Druckkopf, der aus vielen kleinen Nadeln besteht. Zeichen werden dadurch gebildet, daß die jeweiligen Kombinationen von Nadeln gegen das Farbband geschleudert werden.



Der Druckkopf eines Typenraddruckers sieht aus wie ein aufgeblühtes Gänseblümchen. Am Ende jedes „Blütenblattes" sitzt ein Zeichen.

Hinweise zum Laden und Speichern

Programme auf Kassetten laden und speichern hat manchmal seine Tücken, und man kann unter Umständen Stunden damit zubringen, ein Programm, das man gespeichert hatte, in den Computer zu laden. Hier werden einige der Schwierigkeiten beschrieben, auf die man stoßen kann, und Hinweise für deren Abhilfe gegeben.

Programme laden

Um ein Programm von der Kassette mit Erfolg in den Computer zu laden, muß man den Lautstärkeregler (Volume) des Kassettenrecorders richtig einstellen. Ist die Ausgangslautstärke des Recorders zu stark oder zu schwach, dann kann der Computer die Signale nicht aufnehmen.

Die richtige Einstellung zu finden, ist nur eine Sache des Ausprobierens. Hat Ihr Recorder einen Klangregler, dann stellen Sie diesen auf höchste Töne (Treble) ein; die Lautstärke sollte auf eine mittlere Stärke eingestellt sein.

Folgen Sie den Anweisungen zum Laden von Programmen in Ihrem Computer-Handbuch und drükken Sie auf dem Recorder die Wiedergabetaste (PLAY). Die meisten Computer zeigen an, wenn das Programm zwar vollständig geladen worden ist, aber voller Fehler steckt, weil der Recorder die Signale verstümmelt weitergegeben hat.

Wenn der Ladeversuch nicht erfolgreich war, spulen Sie das Band zurück, geben dem Computer NEW ein, damit der Speicher wieder frei wird, und versuchen es dann noch einmal mit etwas veränderter Lautstärkeeinstellung. Nach ein paar Versuchen haben Sie bestimmt die richtige Einstellung für Ihren Computer gefunden.

Wenn Sie immer noch Schwierigkeiten haben, sollten Sie vielleicht einmal die Tonköpfe Ihres Recorders mit Hilfe eines Reinigungsbandes säubern. (Solche Bänder bekommt man in Radiogeschäften.) Wenn auch das nicht hilft, kann der Fehler daran liegen, daß das Programm nicht richtig auf das Band aufgenommen worden ist.

Haben Sie das Programm selbst abgespeichert, dann sehen Sie in der rechten Spalte nach, ob Sie alles richtig gemacht haben. Handelt es sich um ein gekauftes Band, dann bitten Sie am besten einen anderen Computer-Besitzer, es mit seinem Gerät auszuprobieren. Funktioniert es auch bei diesem nicht, sollten Sie das in dem Geschäft reklamieren, in dem das Band gekauft wurde.

Wenn Sie die richtige Lautstärkeeinstellung auf Ihrem Kassettenrecorder gefunden haben, müßten Sie eigentlich alle Programme laden können, die Sie selbst gespeichert haben. Für fremde Bänder kann allerdings eine etwas andere Einstellung erforderlich sein.

Programme speichern

Beim Abspeichern eines Programms brauchen Sie sich nicht um die Einstellung des Lautstärkereglers am Recorder zu kümmern. Folgen Sie einfach den Anweisungen zum Speichern von Programmen in Ihrem Computer-Handbuch und drücken Sie die Aufnahmetaste (RECORD) des Kassettenrecorders. Es ist sinnvoll, dasselbe Programm mehrmals aufzunehmen.

Nach dem Speichern können Sie das Band mit dem Programm wie jedes andere Band abspielen. (Stekken Sie dabei den Kopfhörer aus, damit die Wiedergabe über den Lautsprecher erfolgt!) Nun sollten Sie eine Folge hoher Pieptöne hören. Anschließend versuchen Sie, das Programm in den Computer zu laden, um zu sehen, ob das funktioniert. Ist die Aufnahme schlecht, werden Sie es nicht laden können; oder Sie können es zwar laden, stoßen aber ständig auf Fehler.

Eine schlechte Aufnahme kann daran liegen, daß das Kassettenband alt oder zerknittert ist oder daß die Tonköpfe des Recorders verschmutzt sind. Reinigen Sie die Köpfe mit Hilfe eines Reinigungsbandes und versuchen Sie, das Programm auf ein anderes Band aufzunehmen. Ist das Programm aus dem Speicher des Computers verschwunden, müssen Sie es noch einmal neu eingeben.

Falls Sie das Programm immer noch nicht laden können, liegt das vielleicht daran, daß die Signale Ihres Computers beim Abspeichern für den Kassettenrecorder zu stark oder zu schwach waren. Die meisten einfachen Kassettenrecorder regeln die Signalstärke automatisch; in diesem Fall können Sie die Stärke der vom Computer ausgehenden Signale nicht an den Recorder anpassen. Die einfachste Lösung besteht dann darin, sich einen anderen Kassettenrecorder zu besorgen und auszuprobieren, ob er sich mit Ihrem Computer „versteht". Sie können sich auch bei einem Computer-Klub oder einem Computer-Fachgeschäft Rat holen. Wenn Sie mit Computer und Kassettenrecorder zum Fachhändler gehen, kann er die Geräte überprüfen und eventuell die Kassette oder den Computer so einstellen, daß sie zusammenpassen.

Wenn Ihr Kassettenrecorder einen verstellbaren Pegelregler hat, dann können Sie das Programm mit unterschiedlichen Signalstärken aufnehmen, so lange, bis Sie die richtige Einstellung gefunden haben.

ASCII-Tabelle

Codezahl	ASCII-Zeichen	Codezahl	ASCII-Zeichen
32	Leerstelle	62	>
33	!	63	?
34	"	64	@
35	#	65	A
36	$	66	B
37	%	67	C
38	&	68	D
39	'	69	E
40	(	70	F
41	)	71	G
42	*	72	H
43	+	73	I
44	,	74	J
45	–	75	K
46	.	76	L
47	/	77	M
48	0	78	N
49	1	79	O
50	2	80	P
51	3	81	Q
52	4	82	R
53	5	83	S
54	6	84	T
55	7	85	U
56	8	86	V
57	9	87	W
58	:	88	X
59	;	89	Y
60	<	90	Z
61	=		

ZX81-Code-Tabelle

Codezahl	ZX81-Zeichen	Codezahl	ZX81-Zeichen
11	"	41	D
12	£	42	E
13	$	43	F
14	:	44	G
15	?	45	H
16	(	46	I
17	)	47	J
18	>	48	K
19	<	49	L
20	=	50	M
21	+	51	N
22	–	52	O
23	*	53	P
24	/	54	Q
25	;	55	R
26	,	56	S
27	.	57	T
28	0	58	U
29	1	59	V
30	2	60	W
31	3	61	X
32	4	62	Y
33	5	63	Z
34	6		
35	7		
36	8		
37	9		
38	A		
39	B		
40	C		

Bildschirmgrößen einiger Heimcomputer-Modelle

	Anzahl der Spalten (maximale Anzahl von nebeneinander stehenden Zeichen)	Anzahl der Zeilen (maximale Anzahl der untereinander stehenden Zeichenreihen)
VC-20	22	23
TRS-80 Modell I/III	64	16
Acorn/BBC	20/40/80	16/24/32
Sinclair ZX81	32	22
Sinclair Spectrum	32	22
Apple II	40	24

Tips für den Kauf

Wer sich mit Heimcomputern noch nicht beschäftigt hat, für den ist die Fachsprache sicher ziemlich verwirrend. Beim Kauf des ersten Computers sollte man über einige Dinge Bescheid wissen, deswegen sind hier Erläuterungen zu finden, die helfen sollen, die gebräuchlichen Fachausdrücke besser zu verstehen.
Ein guter Weg, mehr über Computer zu erfahren, ist, Freunde oder Bekannte zu fragen, die bereits einen Computer haben. In den Fachzeitschriften findet man außerdem Marktübersichten und Besprechungen einzelner Computer. Im Fachhandel wird man Fragen stellen können; nützlich ist es auch, sich an einen Computer-Klub zu wenden und dort gezielt nachzufragen. Die Anschriften solcher Klubs werden laufend in den Fachzeitschriften veröffentlicht.
Vor dem Kauf eines Computers sollte man sich darüber im klaren sein, wozu man das Gerät einsetzen und wieviel man dafür ausgeben möchte. Dient es zum Spielen, ist es ein erweiterter Taschenrechner; vielleicht möchte man viele Daten speichern, oder es sollte die Funktion einer Schreibmaschine übernehmen. Will man damit Bilder „malen" oder komponieren, wird auf andere Ausstattungsdetails zu achten sein.
Entscheidet man sich für ein kleines, preisgünstiges Modell, so ist vor dem Kauf zu prüfen, ob der Speicher erweitert werden kann und ob es dazu auch passendes Zubehör gibt, etwa ein Diskettenlaufwerk oder einen Drucker. Wer Fortschritte beim Programmieren und Arbeiten mit dem Computer gemacht hat, dem wird ein ganz einfacher Computer bald nicht mehr genügen, und er wird versuchen, das Gerät zu erweitern.

Prozessor: Der Mikroprozessor ist das „Herz" des Computers. In der technischen Beschreibung wird normalerweise auch der Prozessortyp aufgeführt. Die zur Zeit am weitesten verbreiteten Typen sind der 6502 und der Z80 (siehe Seite 28). Beim Kauf des ersten Computers ist dies jedoch ohne Bedeutung.

Tastatur: Die meisten Mikrocomputer haben eine Tastatur, die der einer Schreibmaschine entspricht. Manche Geräte haben Tasten, die auf Berührung ansprechen, sich dabei aber nicht bewegen. Man muß sich daran gewöhnen, damit längere Programme einzugeben.
Nicht alle Mikrocomputer haben die gleiche Buchstabenfolge, wie sie bei deutschen Schreibmaschinen anzutreffen ist (QWERTZ – die ersten Buchstaben in der oberen Reihe). Im englischen Sprachraum ist QWERTY üblich. Sinclair-Computer haben zum Beispiel eine besondere Tastenbelegung: Jede Taste trägt neben dem Zeichen noch einen Befehl der Programmiersprache. Dann müssen die Befehle nicht Buchstabe für Buchstabe eingetippt werden.

Bildschirmanzeige: Die Anzahl der Zeichen, die der Computer auf dem Bildschirm anzeigt, wird in Spalten (Zahl der nebeneinander stehenden Zeichen) und Zeilen (Zahl der untereinander stehenden Zeichen) angegeben. Normalerweise „rollt" der Text automatisch nach oben weg, wenn eine Zeile vollgeschrieben ist. Dadurch wird für eine neue Zeile Platz geschaffen.

Grafik: Die Bildqualität richtet sich nach der Anzahl der Bildpunkte, die nebeneinander und untereinander auf den Bildschirm passen. Je mehr Punkte dargestellt werden, desto höher und besser ist die sogenannte Auflösung.

Schnittstellen: Die meisten Mikrocomputer haben Schnittstellen für den Anschluß von Fernsehgerät, Monitor oder Kassettenrecorder. Vielfach sind auch Schnittstellen für Drucker, Diskettenlaufwerke, Steuerknüppel und für die Verbindung mit anderen Computern vorhanden. Schnittstellen lassen sich durch Erweiterungsplatinen auch nachrüsten.

Software: Darunter versteht man alle Programme für einen Mikrocomputer, ob sie nun auf Kassette, Diskette oder in gedruckter Form vorliegen. Software, die für einen bestimmten Computertyp entwickelt wurde, läuft häufig nicht ohne weiteres auf anderen Modellen. Ausnahmen machen meist die Geräte des gleichen Herstellers.

Stellvertretend für die vielen auf dem Markt befindlichen Heimcomputer sind hier zwei Modelle näher beschrieben.

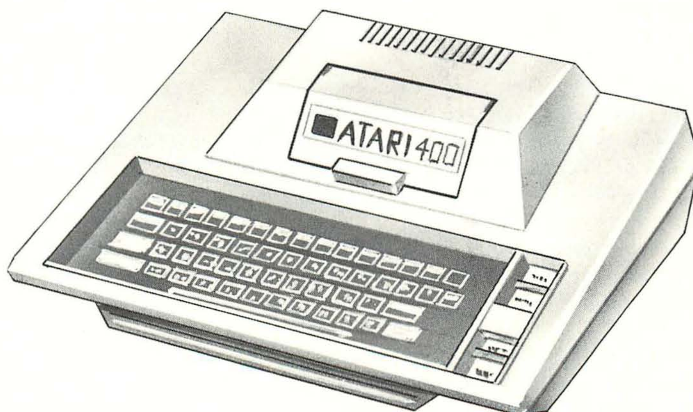
Sinclair ZX Spectrum



Z80A-Prozessor
16 KB Ram, erweiterbar auf 48 KB
Bildschirmanzeige 32 Spalten x 24 Zeilen
Bildschirmauflösung 256 x 192

Mit diesem Modell lassen sich Bilder in acht Farben und Töne (Tonumfang 10 Oktaven) erzeugen. Anschlüsse für ein Farbfernsehgerät und für einen Kassettenrecorder zur Speicherung der Programme sind ebenso vorhanden wie Schnittstellen für einen Drucker und ein Laufwerk, mit dem Programme auf kleinen Magnetplatten gespeichert werden können. Auch eine RS232C- und eine Datennetz-Schnittstelle sind erhältlich.

Atari 400



6502-Prozessor
16 KB Ram, nicht erweiterbar
Bildschirmanzeige 40 Spalten x 24 Zeilen
Bildschirmauflösung 320 x 192

Der Atari 400 hat eine flache QWERTZ-Tastatur. Man kann ein Fernsehgerät anschließen, benötigt aber einen speziellen Recorder zum Laden und Speichern von Programmen. Ein großer Teil der Software liegt jedoch in Modulform vor, die direkt an das Gerät angeschlossen werden kann. Mit Steuerknüppeln kann man viele Spiele spielen.Ein Diskettenlaufwerk und ein Drucker sind als Zubehör erhältlich. Das Gerät bietet eine Grafik-Möglichkeit für 16 Farben und einen Tonkanal.

Wichtige Fachbegriffe

Ablaufdiagramm: Die grafische Darstellung organisatorischer und logischer Ablaufzusammenhänge. Ablaufdiagramme werden zur Entwicklung und Erläuterung von Programmen gebraucht.

Adresse: Den Speicher eines Computers kann man sich in mehrere Segmente aufgeteilt vorstellen. Diese Segmente nennt man Speicherpositionen. Sie sind fortlaufend numeriert; die Nummer der Speicherposition wird als Adresse bezeichnet. Über die Adresse erreicht man den Inhalt der Speicherposition.

ALU: Siehe **Arithmetische und Logische Einheit.**

Animation: Von einem Computer hergestellte, bewegte Bilder auf dem Bildschirm.

Anweisung: Die Anweisung sagt dem Computer, daß er etwas tun soll. Oft wird statt „Anweisung" auch der Begriff „Befehl" gebraucht.

Arbeitsspeicher: Auch „Hauptspeicher" genannt. Hier werden Programme und Daten während der Ausführung des Programms abgelegt.

Arithmetisch-logische Einheit: In den Schaltkreisen der Arithmetisch-logischen Einheit (engl. **A**rithmetic and **L**ogic **U**nit = ALU) wird die gesamte Rechenarbeit des Computers abgewickelt.

ASCII: Engl. **A**merican **S**tandard **C**ode for **I**nformation **I**nterchange (soviel wie Amerikanischer Normcode für Informationsaustausch); eine bestimmte Methode, Buchstaben und Zahlen in Binärzahlen darzustellen.

Assembler: Eine Symbolsprache, die dem Maschinencode sehr nahesteht; er erfordert eine genaue Kenntnis der Funktionen des Computers.

BASIC: Engl. **B**eginners' **A**ll Purpose **S**ymbolic **I**nstruction **C**ode (soviel wie Allzweck-Symbolsprache für Anfänger); eine weitverbreitete, höhere Programmiersprache, die leicht zu erlernen ist und die von den meisten Mikrocomputern verstanden wird.

Binärsystem: Im Binärsystem werden alle Zahlen, Daten und Informationen mit Hilfe der Ziffern 0 und 1 dargestellt. Der Maschinencode verwendet das Binärsystem.

Bit: Kurzform von **bi**nary digi**t**; die kleinste Einheit zur Darstellung von Daten im Binärsystem. Im Computer bedeutet das Bit „kein Impuls" z. B. 0, das Bit „Impuls" 1. Bit ist auch die kleinste Maßeinheit für Informationen.

Bug: Engl. bug = Wanze; ein Fehler im Programm.

Bus: Sammelleitung, an der mehrere Sender bzw. Empfänger (Zentraleinheit, Speicher) angeschlossen sind.

Byte: Ein Byte besteht aus acht Bit und ist eine größere Maßeinheit für Informationen.

Chip: Engl. chip = Scheibchen; eine hauchdünne, winzig kleine Siliziumscheibe, auf der zahlreiche Schaltkreise eingeätzt sind. Verpackt in ein schützendes Gehäuse, enthalten Chips alle Funktionen eines Computers.

COBOL: Engl. **Co**mmon **B**usiness **O**riented **L**anguage; eine in den USA entwickelte Programmiersprache für wirtschaftliche Aufgaben.

Compiler: Ein Übersetzungsprogramm, das Programme, die in einer höheren Programmiersprache geschrieben sind, in den Maschinencode überträgt.

Computer: Ein elektronisch gesteuertes Gerät, das mit einem gespeicherten Programm Daten verarbeiten kann. Ein Computer besteht im wesentlichen aus Ein- und Ausgabe, Verarbeitungswerken (Zentraleinheit) und Speicher(n).

CPU: Siehe **Zentraleinheit**.

Daten: Alle Informationen, die man in den Computer eingibt, ebenso wie Informationen und Ergebnisse, die der Computer entsprechend den Anweisungen in seinem Programm erarbeitet hat und ausgibt.

Datenträger: Mittel zur Speicherung von Daten, z. B. Magnetbänder, -platten oder -trommeln, Lochkarten oder -streifen, Disketten.

Debugging: Siehe **Bug**. Ausmerzen von Fehlern im Programm.

Dialekte: Verschiedene, geringfügig voneinander abweichende Versionen einer Programmiersprache.

Diskette: Biegsame, magnetisierbare Kunststoffplatte (ähnlich einer Single-Schallplatte) zur Speicherung von Daten.

Drucker: Ein Gerät, das Daten auf Papier druckt, die der Computer ausgegeben hat. Auch eine elektronische Schreibmaschine kann ein Drucker sein.

Festwertspeicher: Siehe **ROM**.

Flußdiagramm: Mit Hilfe eines Flußdiagramms (engl. flow chart) kann man die einzelnen Arbeitsschritte eines Programms stufenweise aufbauen.

FORTRAN: Kurzform von **For**mula **Tran**slator (Formelübersetzer); eine höhere Programmiersprache, besonders für technisch-wissenschaftliche Aufgaben.

Gatter: Durch Gatterschaltungen werden die Informationen in binärer Form verarbeitet, indem die Impulse, die durch die Schaltkreise des Computers laufen, von Transistoren gestoppt oder weitergeleitet werden.

Hard copy: Englische Bezeichnung für Computer-Ausdrucke von Daten oder Programmen.

Hardware: Englische Bezeichnung für alles, was zum Computer gehört und sich anfassen läßt.

Hex: Kurzform von Hexadezimal; ein Zahlensystem auf der Basis von 16 (Ziffern 0 bis 9 und Buchstaben A bis F). Ein Byte kann man in zwei Hex-Ziffern schreiben. Dieses System ist vor allem für einfache Programmiersprachen nützlich.

IC: Siehe **Integrierter Schaltkreis**.

Input: Englische Bezeichnung für die Eingabe von Daten in den Computer.

Integrierter Schaltkreis: Engl. **I**ntegrated **C**ircuit = IC; winzige elektrische Schaltkreise. Ein Chip ist gleichbedeutend mit integrierten Schaltkreisen.

Interface: Siehe **Schnittstelle**.

Interpreter: Englische Bezeichnung für Übersetzer. Der Interpreter befindet sich häufig im ROM und übersetzt die Anweisungen aus der Programmiersprache in den Maschinencode, während das Programm läuft.

Kilobyte (KB): 1 Kilobyte = 1024 Bytes.

Kompatibilität: Geräte sind kompatibel, wenn sie ohne Schwierigkeiten miteinander verbunden oder gegeneinander ausgetauscht werden können. In diesem Fall spricht man von Hardware-Kompatibilität.
Als Software-Kompatibilität bezeichnet man die Austauschbarkeit von Programmen. Beim Kauf von Geräten und Programmen sollte man auf die Kompatibilität achten.

Listing: Ein hand- oder maschinengeschriebenes oder auch ausgedrucktes Programm.

Laden: Ein Programm von der Kassette oder Diskette in den Speicher des Computers übertragen.

Magnetplatte: Ein fester, rotierender Datenträger, bei dem die Informationen in den magnetischen Schichten auf der Ober- und Unterseite einer Platte untergebracht werden.

Maschinencode: Ein System binärer Zeichen, die den Computer arbeiten lassen.

Mikroprozessor: Siehe **Zentraleinheit**.

Modem: Kurzform für **Mo**dulator/**Dem**odulator; ein Gerät, das die Maschinencode-Signale aus dem Computer in elektrische Signale umwandelt, die durch Telefonkabel weitergeleitet werden können.

Monitor: Dies ist ein Programm, das oft zusammen mit dem Interpreter im ROM gespeichert ist und der Zentraleinheit die Arbeitsanweisungen gibt, die durch Druck auf die Tastatur ausgelöst worden sind.
Auch anderes Wort für Bildschirm oder Sichtgerät.

Netz(werk): Eine Kombination von Computern und/oder Peripheriegeräten, die untereinander in Verbindung stehen, um Daten oder Programme auszutauschen.

Online: Direkter Anschluß von Geräten an einen Computer.

Output: Englische Bezeichnung für die Ausgabe von Daten oder Arbeitsergebnissen des Computers.

Pascal: Eine höhere Programmiersprache, die vielseitig einsetzbar ist; benannt nach dem französischen Mathematiker Blaise Pascal (1623 – 1662).

PCB: Siehe **Platine**.

Peripheriegeräte: Zusatz- oder Hilfsgeräte, die an den Computer angeschlossen werden können, z. B. Drucker, Plotter oder zusätzliche Bildschirme.

Pixel: Winziger Bildpunkt auf dem Bildschirm. Der Computer läßt solche Bildpunkte aufleuchten und bildet auf diese Weise Buchstaben oder Zeichen.

Platine: Kunststoffbrett mit stromleitenden Kupferstegen, auf dem alle Bauteile des Computers (Chips, Kondensatoren, Widerstände usw.) befestigt sind (engl. **P**rinted **C**ircuit **B**oard = PCB).

Plotter: Rechnergesteuerte Zeichenmaschine, die vom Computer ausgegebene Daten grafisch darstellen kann.

Port: Buchse am Computer, in die Kabelstecker zur Verbindung mit Zusatzgeräten eingesteckt werden können.

Programm: Die Arbeitsanweisung für den Computer. Sie beschreibt in einzelnen Schritten, woher die Daten kommen, wie sie miteinander kombiniert und verarbeitet und wohin sie geleitet werden müssen.

Programmablaufplan: Im Programmablaufplan ist die Reihenfolge festgelegt, in der der Computer die gegebenen Anweisungen auszuführen hat. Damit der Computer die Anweisungen ausführen kann, müssen sie in eine logische Reihenfolge gebracht sein.

Programmiersprache: Für die elektronische Verarbeitung von Daten entwickelte Sprache, mit deren Hilfe Informationen zur Steuerung der Arbeitsabläufe an den Computer übermittelt werden.
Es gibt sogenannte höhere Sprachen, die mit Wörtern und Symbolen arbeiten und für Menschen leichter zu benutzen sind als einfachere Programmiersprachen, die dem Maschinencode näherstehen.

RAM: Engl. **R**andom **A**ccess **M**emory (soviel wie Direktzugriffspeicher). Dies ist der Schreib-/Lesespeicher oder zeitweilige Speicher. Hier wird alles gespeichert, was man in den Computer eingibt. Man kann die dort gespeicherten Daten abrufen („lesen"), verändern oder ergänzen („schreiben"). Dieser Speicher wird in der Regel gelöscht, wenn man den Computer abschaltet.

ROM: Engl. **R**ead **O**nly **M**emory (soviel wie Nur-Lese-Speicher). In diesem Festwert- oder Dauerspeicher sind die Arbeitsanweisungen für den Computer (z. B. auch der Interpreter) niedergelegt. Aus dem ROM kann man lediglich Daten abrufen.

Schnittstelle: Eine Schnittstelle (engl. interface) ermöglicht den Übergang zwischen zwei Geräten; sie besteht oft aus speziellen Schaltkreisen, die die Signale aus dem Computer so umwandeln, daß sie von anderen elektronischen Geräten verarbeitet werden können.
Auch bei der Verbindung von Programmen spricht man von Schnittstellen.

Schreib-/Lesespeicher: Siehe **RAM**.

Software: Englische Bezeichnung für Computer-Programme.

Speicher: Der Teil eines Computers, der Daten aufnimmt, sie jederzeit verfügbar aufbewahrt und auf Abruf abgibt. Der Hauptspeicher befindet sich in der Zentraleinheit des Computers, andere Speicher arbeiten als Peripheriegeräte und sind mit der Zentraleinheit verbunden.

Syntax-Fehler: Rechtschreibfehler in einem Programm (engl. syntax error).

Tastatur: Eingabeeinrichtung eines Computers mit einer Tastatur (engl. keyboard) ähnlich der einer Schreibmaschine. Mit Hilfe der Tastatur kann man Daten in den Computer eingeben, abrufen oder auf dem Bildschirm erscheinen lassen.

Transistor: Elektronisches Bauelement, das elektrische Impulse in den Schaltkreisen des Computers stoppt oder weiterleitet – entsprechend der Impulskombination, die er empfängt. Ein einziger Chip kann Tausende von Transistoren tragen.

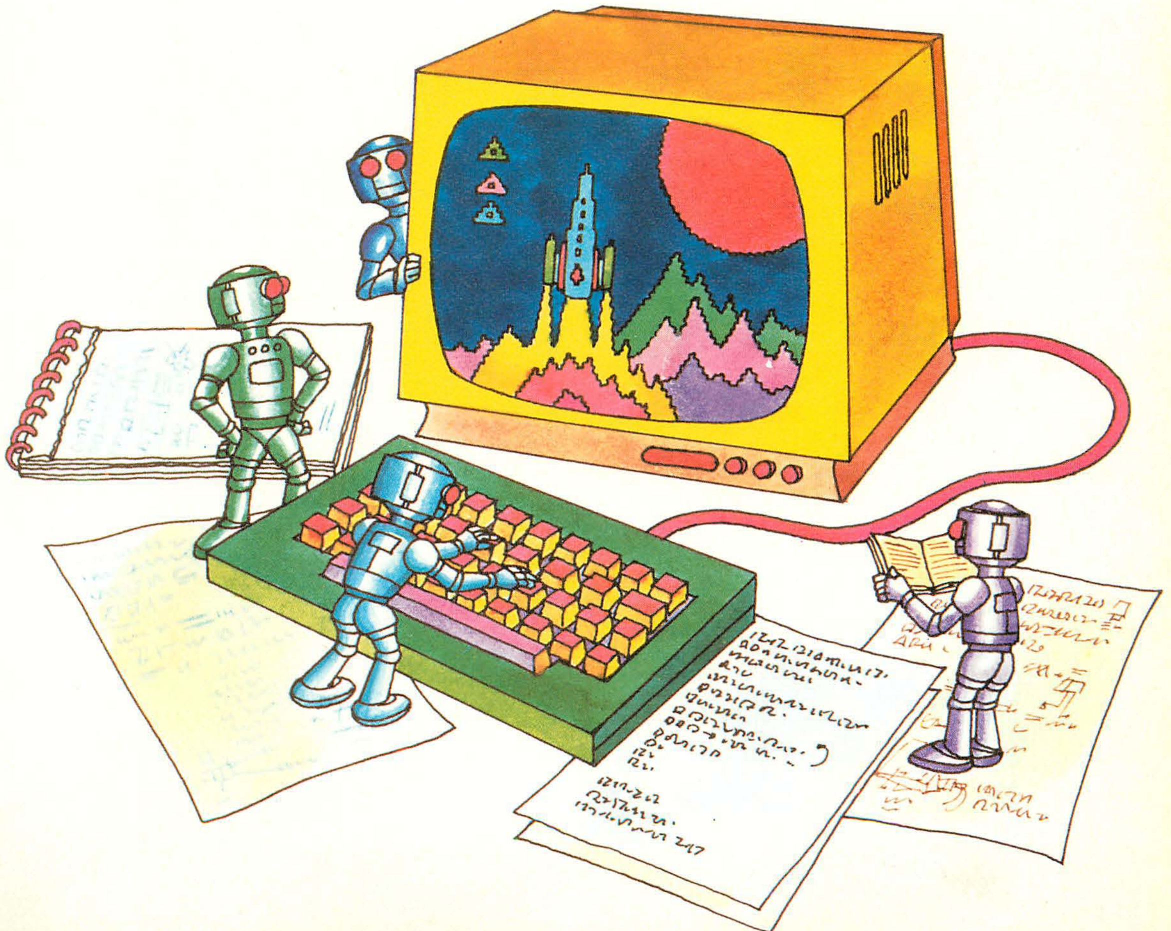
Zentraleinheit: Die Zentraleinheit enthält das Steuer- und Rechenwerk (auch Mikroprozessor, engl. **C**entral **P**rocessing **U**nit = CPU) sowie den Hauptspeicher. In der Zentraleinheit übernimmt das Steuerwerk die Überwachung und Steuerung beim Ablauf der Programme, das Rechenwerk kombiniert und verarbeitet die Daten, und der Hauptspeicher nimmt Daten und Programme zur Verarbeitung auf.

Zweiter Teil

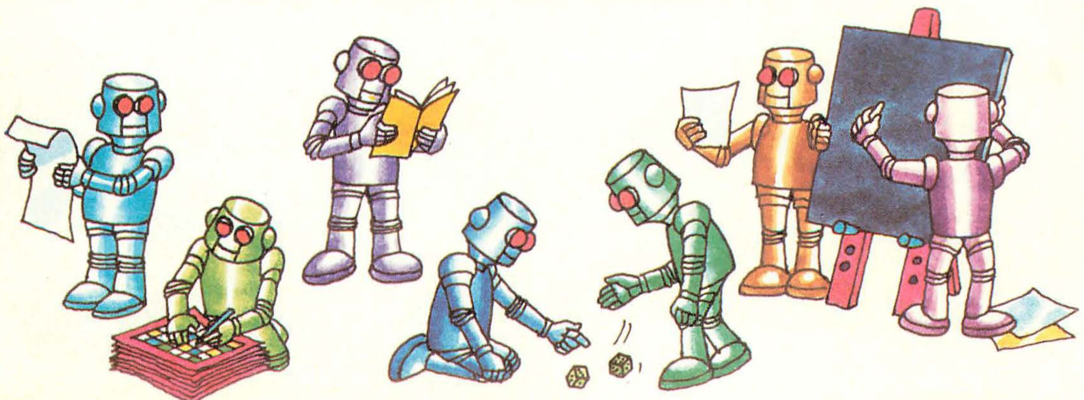
Programmieren ganz einfach

Einstieg mit BASIC

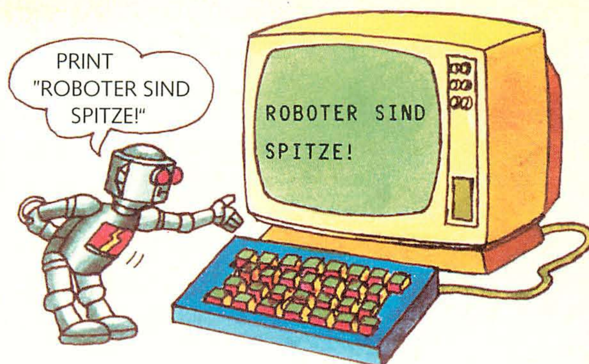
Von Brian Reffin Smith

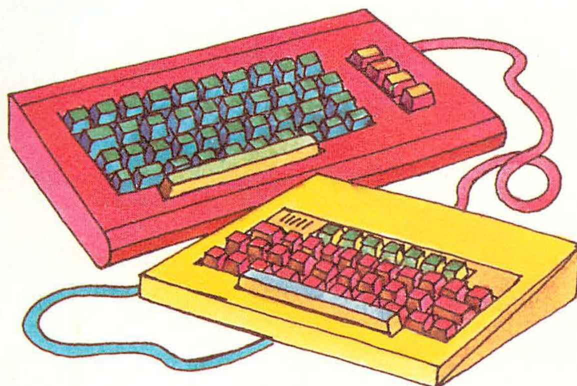


Programmieren – ganz einfach

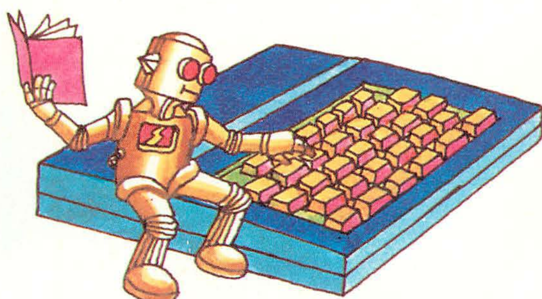


Im zweiten Teil dieses Buches können Sie lernen, Computerprogramme in BASIC selbst zu schreiben. BASIC ist eine Programmiersprache, die von den meisten Heim- oder Homecomputern verstanden wird. Die Anweisungen werden dazu in einer Form geschrieben, die der Computer verstehen kann.



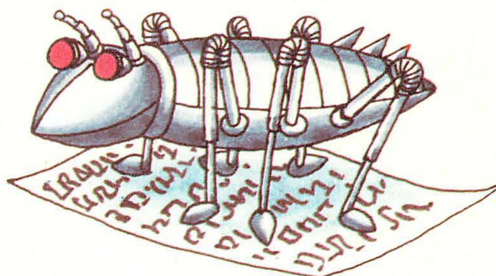


Um BASIC zu lernen, müssen Sie nicht unbedingt selbst einen Computer besitzen. Sie werden die Programme allerdings besser verstehen, wenn Sie diese mit einem Computer ausprobieren können. Die verschiedenen Computertypen verwenden geringfügig unterschiedliche BASIC-Dialekte. Die meisten hier erläuterten Befehle arbeiten auf allen Mikrocomputern; die wenigen Ausnahmen davon sind angegeben.



Zunächst werden einige Hinweise zum Programmieren eines Computers gegeben. Dann werden die wichtigsten BASIC-Befehle schrittweise erklärt. Kurze Programme verdeutlichen, wie sie funktionieren.

Damit Sie das Programmieren gut üben können, finden Sie hier Programmierübungen, Rätsel und praktische Änderungsvorschläge für die aufgeführten Programme. Die Lösungen für Programmierübungen und Rätsel stehen auf Seite 90/91.
Am Schluß dieses Teils sind die wichtigsten BASIC-Befehle und Fachbegriffe zusammengestellt und kurz erläutert. Auch Tips und Kniffe fürs Programmieren fehlen nicht, so zum Beispiel eine kleine Liste von Programmfehlern („Bugs", von engl. bug = Wanze), die den Ablauf eines Programms vorzeitig stoppen.



Wenn Sie einen Computer besitzen, sollten Sie die Programme in diesem Buch ausprobieren. Wenn Sie dann mehr über die Arbeitsweise Ihres Computers wissen wollen, schlagen Sie die BASIC-Befehle im Handbuch Ihres Computers nach.
Unter Umständen müssen einige der hier erläuterten Regeln bei Ihrem Computer nicht befolgt werden. Am besten lernen Sie BASIC, wenn Sie eine Vielzahl von Programmen aus Büchern oder Zeitschriften ausprobieren und sie dann geringfügig abändern. Bald werden Sie eigene Programme schreiben können.

Wie ein Computer arbeitet

Mit einem Computer können Sie die verschiedenartigsten Dinge machen: Sie können Bilder zeichnen, lustige Gedichte schreiben, Schaubilder darstellen, spielen und mit Wörtern und Zahlen arbeiten – zur Arbeitserleichterung oder nur zum Spaß. Man sagt, der Computer „verarbeitet Informationen". Seine Aufgabe ist es, die von Ihnen eingegebenen Informationen zu speichern, sie gemäß Ihren Anweisungen zu verarbeiten und dann das Ergebnis zu zeigen.

TEILE 999 DURCH 27

DRUCKE 80MAL MEINEN NAMEN

ZEICHNE 1000,100

LADE "RAKETENSTART"

POKE 36876,15

Damit der Computer das tut, was Sie wollen, müssen Sie ihm ganz genaue *Anweisungen* oder *Befehle* geben. Eine Folge von Befehlen nennt man *Programm*; die Informationen, die Sie dem Computer geben, sind die *Daten*. Das Programm muß in einer Sprache geschrieben sein, die der Computer verstehen kann, zum Beispiel in BASIC, und es muß den Regeln dieser Sprache entsprechen.



Mikrocomputer

Die meisten Homecomputer bestehen aus einer *Tastatur*, die mit dem Bildschirm eines Fernsehgeräts verbunden ist. Die Anweisungen und Informationen werden über eine Tastatur eingegeben. Alle Eingaben und die vom Computer erstellten Ergebnisse werden auf dem Bildschirm gezeigt.

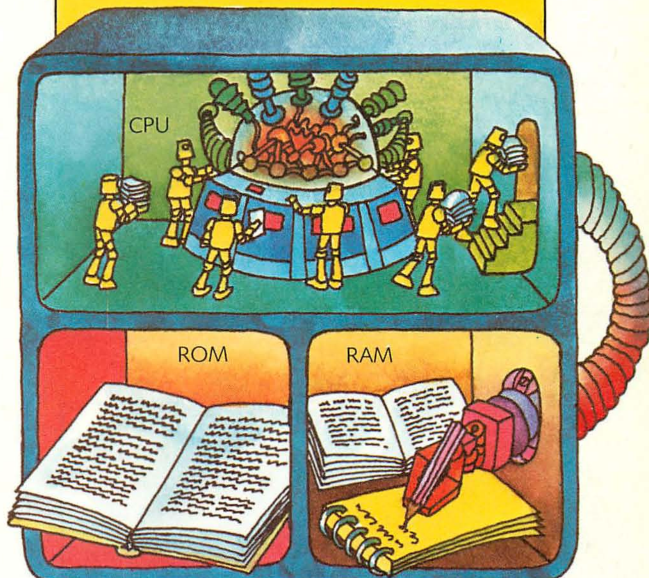
Manche Computer haben eine Flüssigkristallanzeige, wie ein Taschenrechner. Andere haben einen *Monitor*, einen Bildschirm, der im Gegensatz zu einem Fernsehgerät keine Signale eines Senders empfangen kann.

Das Tastenfeld (oder die Tastatur) entspricht dem einer Schreibmaschine, es hat aber noch zusätzliche Tasten. Bei manchen Computern kann mit einer Taste ein ganzer BASIC-Befehl eingegeben werden; er muß dann nicht Buchstabe für Buchstabe getippt werden.



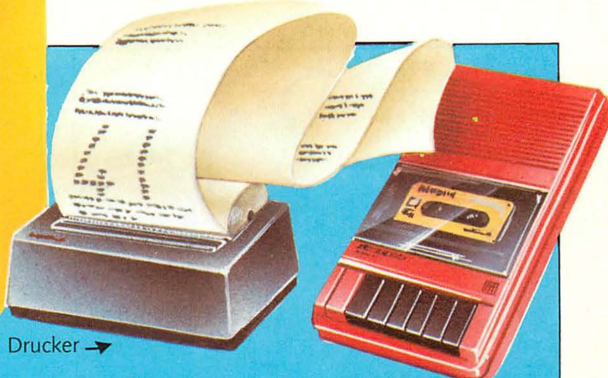
Das Innenleben des Computers

Ein Computer hat zwei wichtige Funktionseinheiten: Der *Mikroprozessor* oder die CPU (engl. **C**entral **P**rocessing **U**nit) ist das Steuer- und Rechenwerk. Im *Speicher* werden Programme und Daten abgelegt.



Genauer gesagt hat der Computer zwei Speicher: Im *Festwertspeicher*, dem ROM (engl. **R**ead **O**nly **M**emory), ist das Programm für die Funktionen des Mikrocomputers abgespeichert. Der *Schreib-/Lesespeicher*, das RAM (engl. **R**andom **A**ccess **M**emory), wird zur vorübergehenden Speicherung von Programmen, Befehlen und Daten benötigt, die über die Tastatur eingegeben werden. Dieser Speicher wird beim Abschalten des Computers gelöscht.



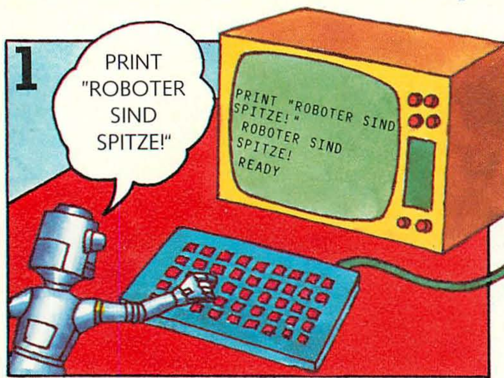


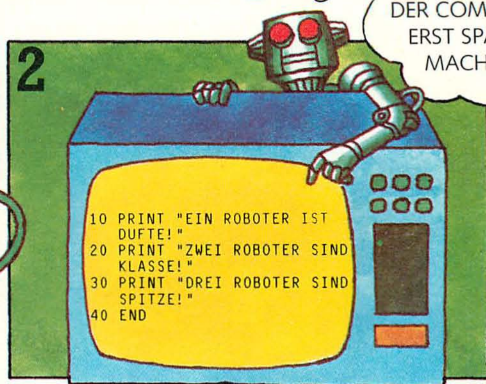
Drucker →

Die Informationen werden in der Regel auf dem Bildschirm ausgegeben. Sie können auch auf einem Drucker ausgegeben werden. Dies ist nützlich, da sonst die Informationen beim Abschalten der Geräte verlorengehen.

Die Informationen können auch mit einem Kassettenrecorder gespeichert werden. Programme und Daten werden auf Magnetband gespeichert und können bei Bedarf wieder in den Computer eingegeben werden.

Wie man einem Computer Befehle gibt

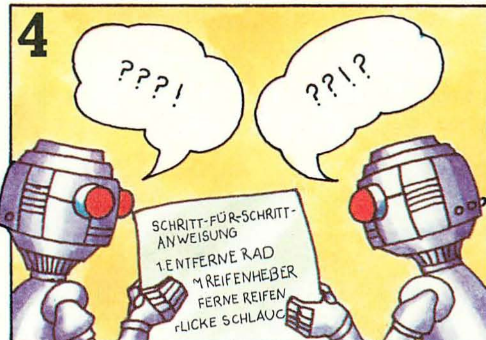




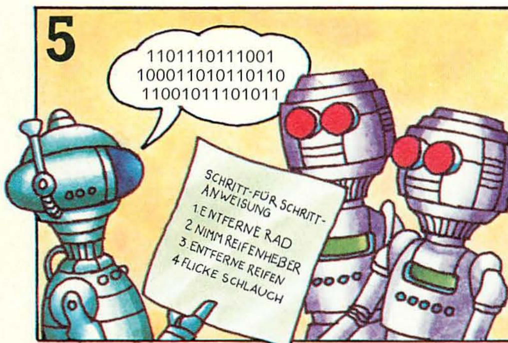
Damit ein Computer etwas ausführt, muß man ihm Befehle in einer Sprache geben, die er versteht. Dies kann ein Befehl sein, der sofort ausgeführt wird, oder ein sogenanntes Programm, eine Folge von Befehlen, die zunächst gespeichert und erst nach dem Start-Befehl ausgeführt werden.



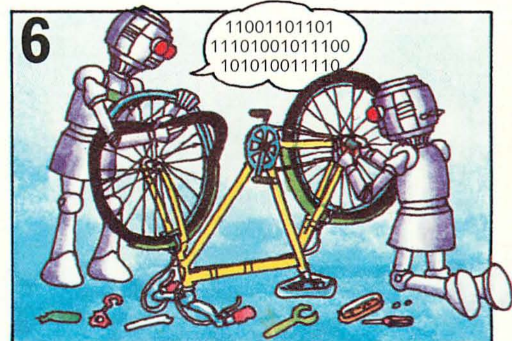
Die Befehle müssen sehr sorgfältig erstellt werden. Der Computer versucht stur, sie auszuführen, auch wenn sie falsch sind.



Er versteht keine Befehle in unserer Sprache, daher müssen sie in einer Computersprache geschrieben werden. Auf der folgenden Seite sind einige Programmiersprachen beschrieben.



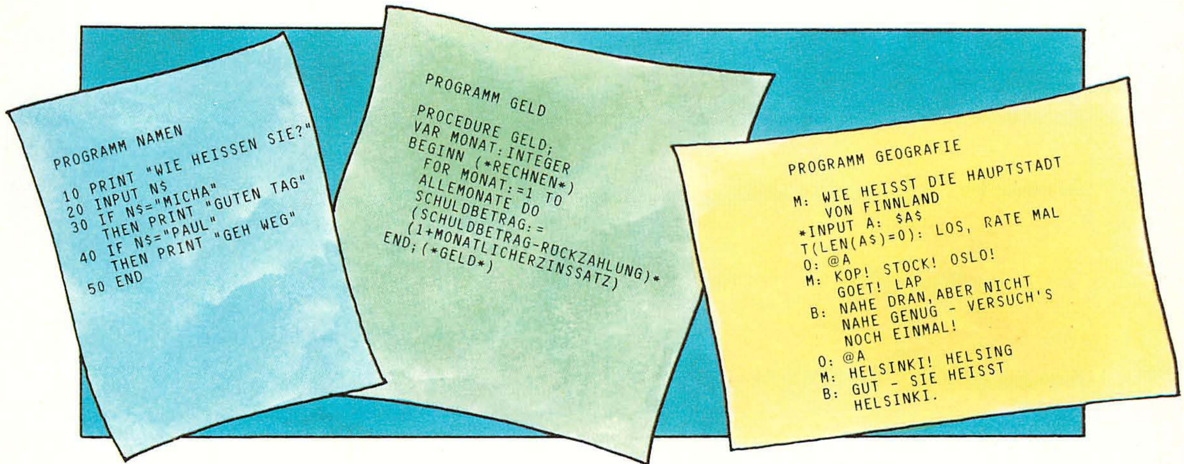
Der Computer arbeitet aufgrund von winzigen Stromimpulsen. Die Befehle werden durch ein spezielles internes Programm in die Maschinensprache, den sogenannten Maschinencode, umgesetzt.



Im Maschinencode wird jede Information durch ein Muster von Impulsen dargestellt. Dieser Code kann mit den Ziffern 1 (Impuls) und 0 (kein Impuls) dargestellt werden.

Computersprachen

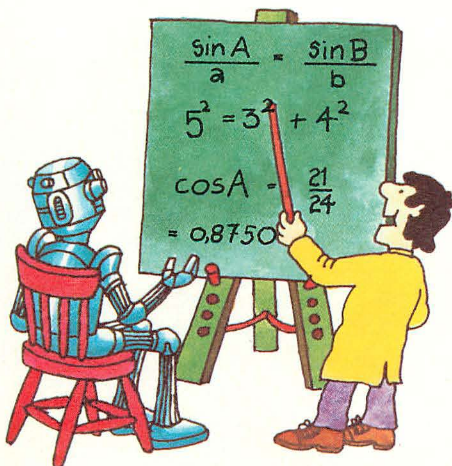
Programme können im Maschinencode geschrieben werden, dies ist aber sehr umständlich. Deswegen wurden spezielle, sogenannte höhere Computersprachen entwickelt, die der Computer in seinen Maschinencode übersetzt.
Es gibt eine Vielzahl von höheren Computersprachen, zum Teil wurden sie nur für einen ganz bestimmten Anwendungszweck entwickelt. BASIC ist eine der weitestverbreiteten Sprachen. BASIC bedeutet **B**eginners' **A**ll purpose **S**ymbolic **I**nstruction **C**ode, also Allzweck-Symbolsprache für Anfänger. Doch nicht nur Anfänger verwenden BASIC. Hier drei Beispiele für verschiedene Computersprachen.

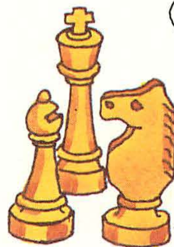


Dies ist ein kleines Programm in BASIC. Zeile 20 befiehlt dem Computer, „Wie heißen Sie?" auf den Bildschirm zu schreiben. Der Computer speichert Ihre Antwort und übermittelt Ihnen eine Botschaft, falls Sie z. B. Hans oder Micha heißen.

Diese Sprache heißt Pascal, zu Ehren des französischen Mathematikers Blaise Pascal (1623 – 1662). Der Ausschnitt stammt aus einem Programm, das Finanzprobleme löst. Viele Leute meinen, gute und klare Programme seien in Pascal leichter zu schreiben als in BASIC.

Diese Sprache heißt PILOT. Sie wird für Unterrichts- und Lernprogramme benutzt. Der Computer kann in dieser Sprache Antworten auch dann erkennen, wenn sie nicht ganz korrekt sind.





11. Bb 3, Ne 5
12. 0-0-0, Nc 4
13. Bxc 4, Rxc 4
14. h 5, Nxh 5



Zunächst erscheinen Computersprachen seltsam und schwierig, wie das Finnisch in unserem Beispiel rechts. Es gibt eine Vielzahl von Problemen, bei denen spezielle Sprachen angewandt werden.

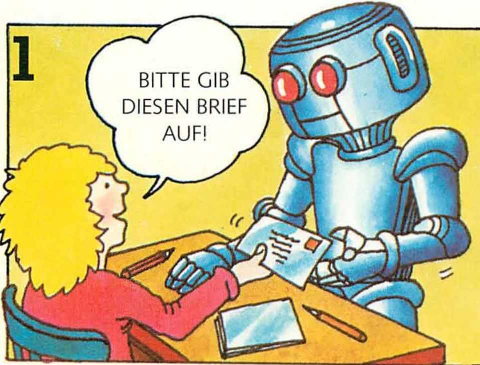
Mathematische Formeln werden zum Beispiel mit speziellen Zeichen und Begriffen geschrieben, auch für Schachzüge oder für Musik werden ganz bestimmte Begriffe und Zeichen verwendet.

* Ich schätze, so minus 15 Grad.

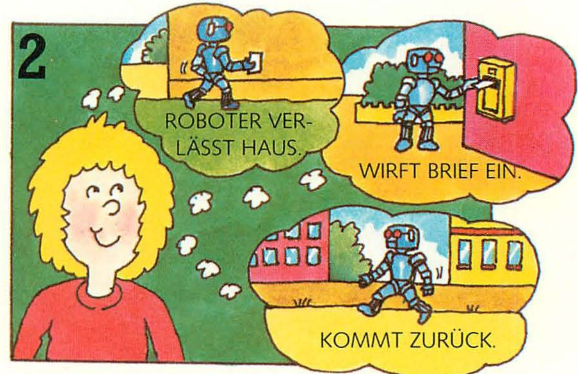
Wie man ein Programm schreibt

Ein Computerprogramm ist wie eine Spielregel oder ein Kochrezept. Sind diese falsch, kann man das Spiel nicht richtig spielen, oder der Kuchen mißlingt. Ebenso sind die Ergebnisse, die ein Computer liefert, von den Anweisungen abhängig, die Sie ihm geben. Bevor Sie ein Programm schreiben, müssen Sie sorgfältig den Ablauf strukturieren und die einzelnen Schritte untersuchen, die zum Erreichen des Ziels notwendig sind.

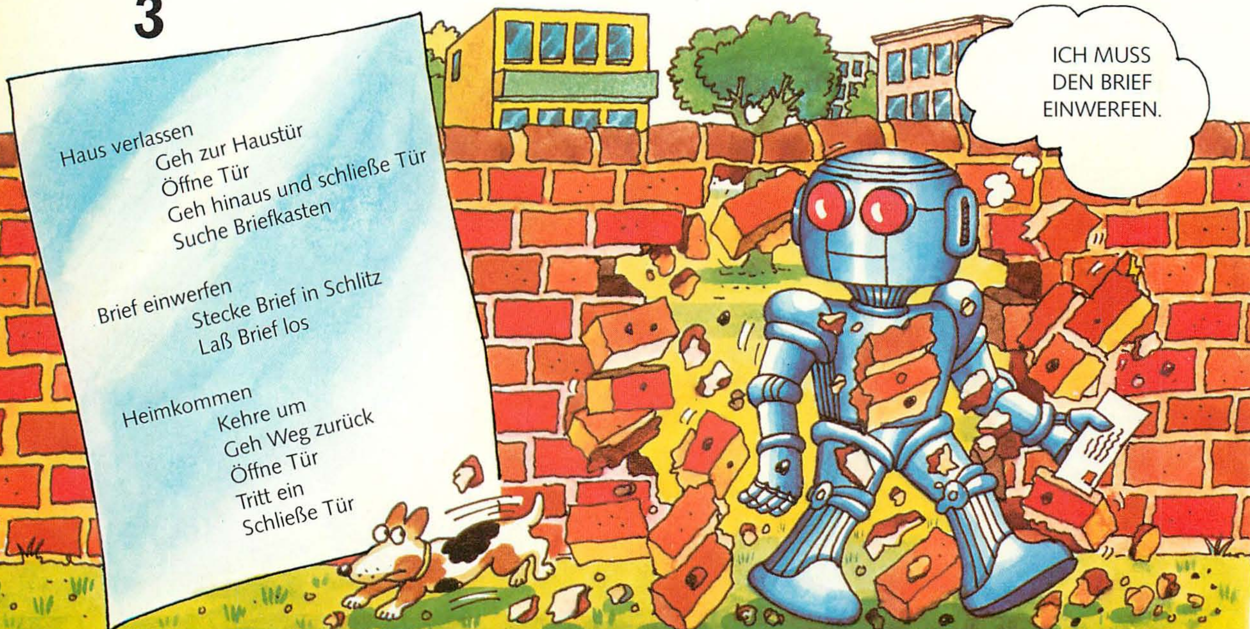
Programm "Brief aufgeben"



Versuchen Sie, ein Programm zu entwickeln, das einen Roboter veranlaßt, einen Brief aufzugeben. Die einfache Anweisung, wie sie in diesem Bild gegeben wird, kann der Computer nicht verstehen.



Sie müssen genau ausarbeiten, was der Computer zu tun hat, um den Brief aufzugeben. Dabei muß dem Computer jeder Schritt einzeln vorgegeben werden.

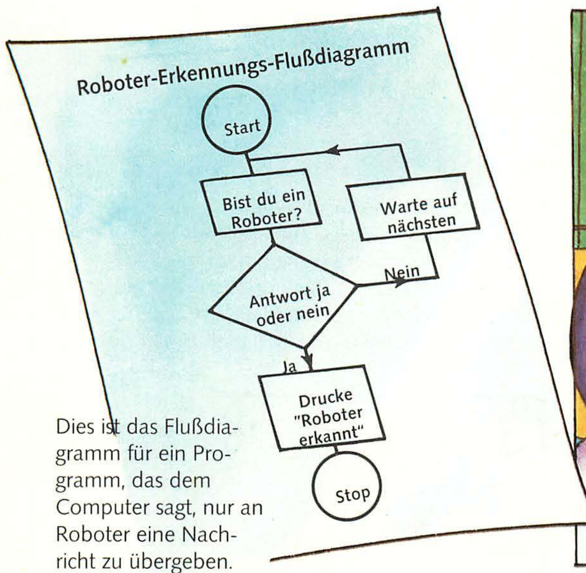


Um das Programm zu schreiben, müssen Sie die Anweisungen in einzelne Schritte zerlegen. Diese Schrittfolge muß in eine Sprache übersetzt werden, die der Computer versteht.

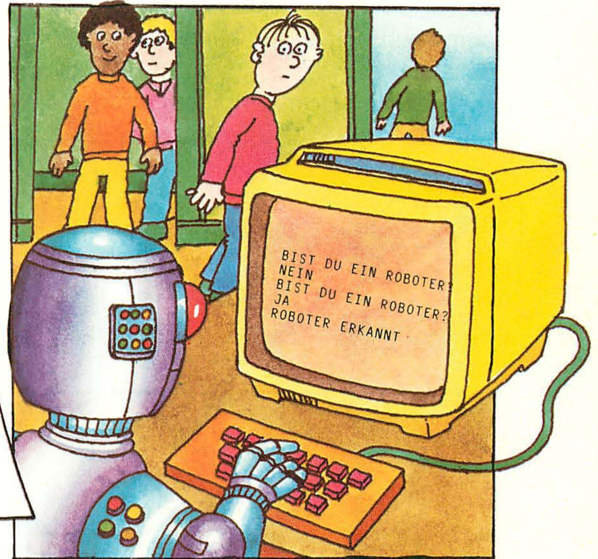
Der Roboter versucht, Ihren Anweisungen zu folgen, auch dann, wenn sie unvollständig oder falsch sind. Die Fehler in einem Programm nennt man „Wanzen" (engl. bug = Wanze); sie führen manchmal zu unerwarteten Ergebnissen – wie das Bild zeigt.

Flußdiagramme

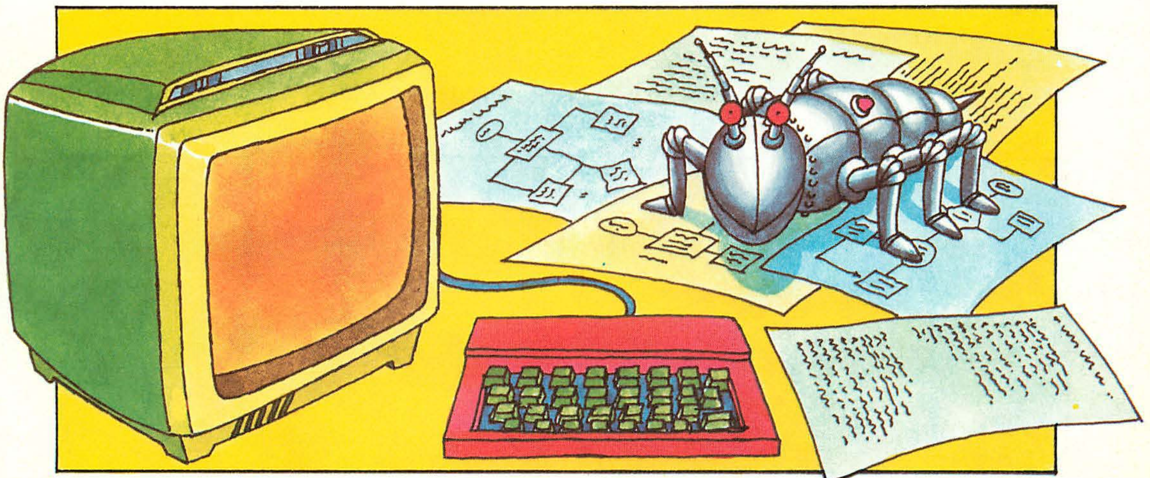
Bevor man ein Programm schreibt, ist es oft hilfreich, sich ein Diagramm zu zeichnen, das die wesentlichen Schritte zur Lösung einer Aufgabe darstellt. Ein solches Diagramm heißt *Flußdiagramm*. Es zeigt jeden Schritt, den der Computer ausführen muß, und die richtige Reihenfolge dieser Schritte.



Dies ist das Flußdiagramm für ein Programm, das dem Computer sagt, nur an Roboter eine Nachricht zu übergeben.



Ein Flußdiagramm besteht aus Feldern, die für die verschiedenen Programmschritte unterschiedlich aussehen. Start und Ende eines Programms haben runde Felder, die Anweisungen an den Computer stehen in rechteckigen Feldern, und die Entscheidungsfelder, bei denen der Computer sich entsprechend der erhaltenen Information entscheiden muß, haben die Form einer Raute. Die Linien zeigen die verschiedenen Wege, denen der Computer folgen kann.



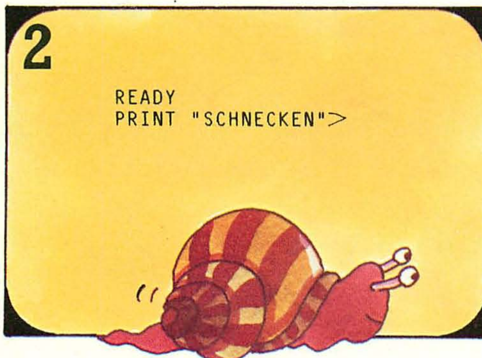
Nachdem alle Einzelheiten des Programms gut überlegt sind, kann man das Programm in BASIC übersetzen und dann auf dem Computer testen. Wahrscheinlich wird dieses Programm nicht sofort laufen, weil es noch einige Fehler enthält. Diese Fehler können Tippfehler sein, die beim Eingeben des Programms in den Computer entstanden sind; es können auch logische Fehler im Programm sein (siehe dazu Seite 88/89). Manchmal verursacht ein Fehler ein etwas anderes Ergebnis, das vielleicht sogar günstig ist. Solche nützlichen Fehler heißen „Möpse" (engl.: pugs).

Erste Schritte in BASIC

Die meisten BASIC-Ausdrücke sind Wörter in englischer Sprache. Mit einigen Englisch-Kenntnissen kann man leicht erraten, was sie bedeuten. So bedeutet z. B. PRINT: „Zeige auf dem Bildschirm", RUN: „Führe das Programm aus" und INPUT: „Gib dem Computer Information". Auf diesen beiden Seiten wird der Gebrauch von PRINT erklärt. In den meisten Heimcomputern ist ein Interpreter für BASIC bereits eingebaut. Wenn man ein solches Gerät einschaltet, kann man es sofort in BASIC programmieren. Bei manchen Computern muß man allerdings erst ein spezielles Programm mittels Kassettenband eingeben, bevor sie BASIC verstehen.



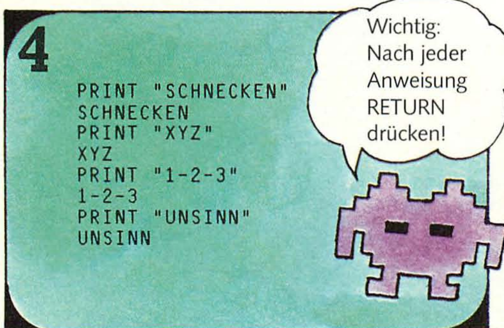
Wenn man den Mikrocomputer einschaltet, werden normalerweise automatisch einige Wörter auf dem Bildschirm erscheinen, zusammen mit einem kleinen Zeichen, das *Cursor* genannt wird. Der Cursor zeigt, an welcher Stelle der nächste Buchstabe erscheinen wird, den man eintippt.



Der Computer läßt Wörter auf dem Bildschirm erscheinen, wenn man PRINT und anschließend in Anführungszeichen den gewünschten Text eingibt. So bedeutet z. B. PRINT "SCHNECKEN", daß der Computer das Wort SCHNECKEN auf dem Bildschirm erscheinen lassen soll.



Der Computer führt Anweisungen erst dann aus, wenn man NEWLINE (RETURN oder ENTER – dies ist von Computer zu Computer unterschiedlich) eingibt.



Der Computer läßt alles das auf dem Bildschirm erscheinen, was man – in Anführungszeichen eingeschlossen – eingibt. Dies können sowohl Buchstaben als auch Zahlen, Wörter oder Zeichen sein. Die Anführungszeichen selbst erscheinen nicht.

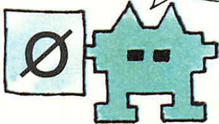


Wenn man nur Zahlen auf dem Bildschirm zeigen will, braucht man keine Anführungszeichen zu benutzen. Der Bildschirm wird wieder frei, wenn man CLS eingibt. (Bitte im Handbuch für Ihren Computer nachprüfen!)

Ein Programm in BASIC

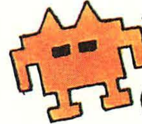
In einem Programm beginnt jede Programmzeile mit einer Nummer. Dies veranlaßt den Computer, die Informationen in seinem Speicher abzulegen und sie nicht eher auszuführen, bis man ihn dazu auffordert. Auf der gegenüberliegenden Seite hatten die Anweisungen an den Computer keine Nummern, daher wurden sie sofort ausgeführt. Hier ist ein kleines Programm, das Zeichen in Form eines Gesichts auf dem Bildschirm erscheinen läßt.

Bei manchen Computern verläuft durch die Zahl 0 ein Schrägstrich, wie hier.




```
10 PRINT "//////"
20 PRINT "I    I"
30 PRINT "I(..)I"
40 PRINT "I -L I"
50 PRINT "VVVVVV"
60 END
```

Die Zeilennummern sind meistens Zehnerzahlen. Man kann dann leicht weitere Programmzeilen hinzufügen, ohne alle Zeilen umnumerieren zu müssen.



Viele Computer benötigen diese Zeile nicht.

Wenn man ein Programm eingibt, muß man RETURN (oder das entsprechende Wort) am Ende jeder Zeile drücken. Die Zeilen erscheinen auf dem Bildschirm, der Computer führt die Anweisungen jedoch erst dann aus, wenn man RUN eingibt.

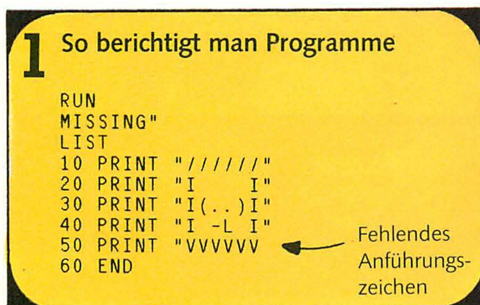
Achtung: Nicht den Buchstaben O mit der Zahl 0 (Null) verwechseln, dies führt zu Fehlern! Die meisten Mikrocomputer haben eine DELETE-(Lösch-)Taste, mit der man Tippfehler verbessern kann.

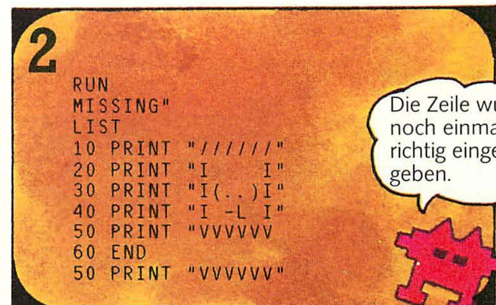




Sobald alle Zeilen eingegeben sind, muß sorgfältig geprüft werden, ob keine Fehler mehr vorhanden sind. Der Computer bearbeitet das Programm, wenn man erst RUN und dann RETURN eingibt.

Wenn das Programm nicht läuft oder das Bild nicht richtig aussieht, muß man das Programm nochmals anschauen und auf Fehler überprüfen. Dazu LIST eingeben. Der Computer gibt dann vielleicht eine Fehlermeldung.





Bei den meisten Fehlern gibt der Computer eine Fehlermeldung. Die Fehlermeldungen werden im Computer-Handbuch erklärt. Am einfachsten korrigiert man einen Fehler, indem man die ganze Zeile nochmals eingibt. Der Computer ersetzt dann die alte Zeile durch die neue. Um eine Zeile ganz aus dem Programm zu entfernen, muß man die Zeilennummer und dann RETURN eingeben. Die Möglichkeiten, Zeilen zu korrigieren oder zu verändern, sind von Computer zu Computer unterschiedlich; dazu verwendet man EDIT oder COPY, entsprechend den Erläuterungen im Computer-Handbuch.



Programmierübung: Versuchen Sie, das Programm abzuändern, um dem Gesicht unterschiedliche Ausdrücke zu geben.

Informationen für den Computer

Der Computer kann natürlich viel nützlichere Dinge tun, als lediglich Figuren auf dem Bildschirm erscheinen zu lassen. Dazu ist es allerdings erforderlich, daß er Informationen oder *Daten* erhält, die er bearbeiten kann. Der Computer speichert diese Informationen in seinem Speicher, bis er Anweisungen erhält, die Informationen zu bearbeiten.

1

```
10 LET A=6
20 LET B=7
30 LET C=23
40 LET D=4
```

Diese Zahlen soll der Computer speichern.

Dies sind die Namen für die Speicherplätze.

In den Speicher des Computers eingegebene Daten müssen eine Bezeichnung bekommen, damit man sie wiederfindet. Dazu kann man Buchstaben verwenden. Um eine Zahl in einen bestimmten Speicherplatz einzugeben, benutzt man das Wort LET, wie oben gezeigt. Ein gekennzeichneter Speicherplatz heißt *Variable*, da er innerhalb des Programms verschiedene Daten zu verschiedenen Zeiten enthalten kann.

2

```
10 LET A=3
20 LET A$="SCHNECKEN"
30 LET B=43
40 LET B$="ROBOTER"
```

Die Anführungszeichen nicht vergessen!

Zum Speichern von Buchstaben und Symbolen in den Speicherplätzen benutzt man eine andere Bezeichnung. Buchstaben und Symbole heißen *Zeichenketten* (engl. strings). Um sie zu bezeichnen, benutzt man die Buchstaben des Alphabets zusammen mit dem Dollar-Zeichen, z. B. C$ (gesprochen „C Dollar").

Zur Speicherung eines Wortes im Speicherplatz wird LET in derselben Weise angewandt wie bei Zahlen-Variablen. Die Buchstaben und Symbole müssen dabei, wie oben gezeigt, in Anführungszeichen stehen.

3

```
10 LET B=365
20 LET T$= "TAGE IM JAHR"
30 LET L$="AUSSER IM SCHALTJAHR"
40 PRINT B
50 PRINT T$
60 PRINT L$
70 END
```

Viele Computer benötigen END nicht.

4

```
RUN
365
TAGE IM JAHR
AUSSER IM SCHALTJAHR
```

Zur Angabe der Information auf dem Bildschirm wird das Wort PRINT zusammen mit dem Namen der Variablen eingegeben, z. B. PRINT A$. Das kurze Programm oben druckt die Information der Variablen B, T$ und L$ aus.

Man kann das Programm beliebig oft laufen lassen – der Computer druckt jedesmal die gleiche Information aus. Die Daten in den Variablen bleiben so lange gleich, bis man sie ändert.

Eine andere Möglichkeit, Informationen zu speichern

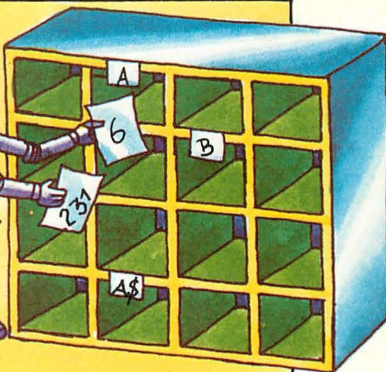
```
10 READ A
20 READ B
30 READ A$
40 DATA 6,231,FREITAG
```

Die Bezeichnungen für Zahlen und Buchstaben müssen richtig gewählt sein.

Kommas

Bei manchen Computern müssen Daten-Wörter in Anführungszeichen eingegeben werden.





Eine andere Möglichkeit der Informationsspeicherung besteht mit den Wörtern READ und DATA, wie oben gezeigt. Die READ-Zeilen bewirken, daß der Computer Speicherplatz benennt, die DATA-Zeile enthält die Information.

Wenn das Programm läuft, legt der Computer die Daten der Reihe nach in einem Speicherplatz ab. Die einzelnen Daten müssen durch Kommas voneinander getrennt sein, damit der Computer sie erkennen kann. (Diese Methode funktioniert nicht auf dem ZX81-Computer.)

Programme

1

```
10 READ Q
20 READ X$
30 DATA 24,HAMBURGER
40 PRINT Q
50 PRINT X$
60 END
RUN
24
HAMBURGER
```

Komma

Das ist ein „Datum".

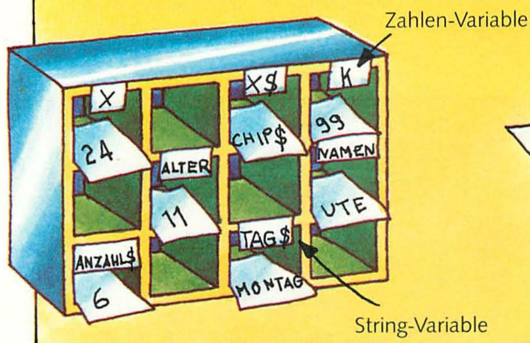
2

Anführungszeichen

```
10 LET A$="ROBOTER SIND NÜTZLICH"
20 LET B$="SIE SIND"
30 LET C$="NÜTZLICHE BLECH-IDIOTEN"
40 PRINT A$
50 PRINT B$
60 PRINT C$
70 END
RUN
ROBOTER SIND NÜTZLICH
SIE SIND
NÜTZLICHE BLECH-IDIOTEN
```

Hier sind zwei Programme. Eines verwendet READ und DATA, das andere LET, um Informationen im Speicher des Computers abzulegen.

Mehr über Variablen

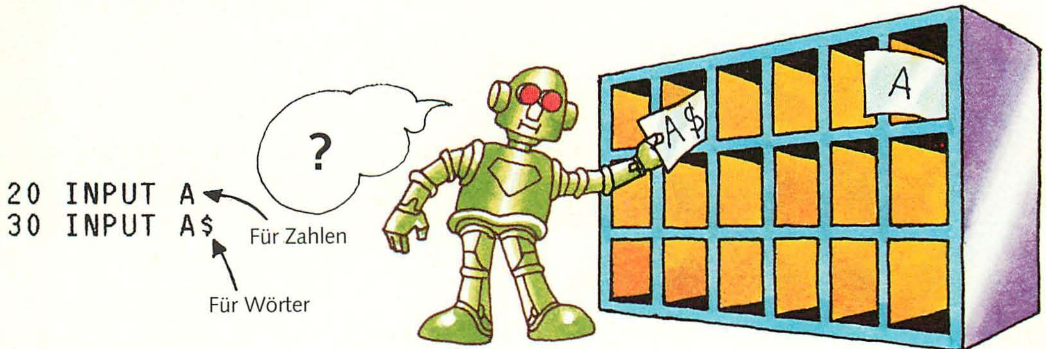




Variablen sind benannte Plätze im Speicher des Computers, in denen Informationen gespeichert werden. Eine Variable, die Zahlen enthält, heißt *Zahlen-Variable*. Eine Variable, die Buchstaben oder Zeichen enthält, heißt *String-Variable*. Der Inhalt einer Variablen kann sich während des Programms ändern. Manche Computer verwenden Wörter als Bezeichnungen für Variablen, aber nicht solche Wörter, die in BASIC vorkommen; dies würde den Computer verwirren.

INPUT

Eine weitere Möglichkeit, Daten in den Computer einzugeben, bietet das Wort INPUT. Es ermöglicht die Informationseingabe während des Programmablaufs. So kann man bei jedem Programmablauf verschiedene Daten benutzen.



INPUT verwendet man mit einem Namen, z. B. A für eine Zahl oder A$ für eine Zeichenkette. Stößt der Computer in einem Programm auf das Wort INPUT, belegt er einen Speicherplatz mit dem jeweiligen Namen und fragt nach den Daten – meist durch Angabe eines Fragezeichens oder eines anderen Symbols auf dem Bildschirm. Dann gibt man die Daten ein, der Computer speichert sie auf dem Speicherplatz und fährt mit dem restlichen Programm fort.



Bild 2 zeigt das Ergebnis dieses Programms. Sobald der Computer das Wort INPUT in Zeile 10 liest, gibt er ein Fragezeichen auf dem Bildschirm aus und wartet auf die Eingabe einer Zahl für G. Dann erscheint ein weiteres Fragezeichen im Anschluß an die INPUT-Anweisung in Zeile 20. Diesmal muß man Wörter oder Symbole eingeben; denn der Name B$ läßt den Computer eine Zeichenkette erwarten.



Wenn Sie einen Computer haben, versuchen Sie, dieses Programm einzugeben. Drücken Sie RUN, um das Programm zu starten. Erbittet der Computer Informationen, dann geben Sie Ihren Namen und Ihr Alter ein oder die entsprechenden Angaben eines Familienmitglieds. Lassen Sie dieses Programm öfter laufen, immer wieder mit anderen Daten. Um es zu starten, muß man jedesmal RUN eingeben. Der Computer gibt immer genau das aus, was man für N$ und A eingegeben hat.

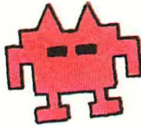
Verse schmieden mit dem Computer

Ihr BASIC reicht jetzt bereits, um mit dem Computer ein „Gedicht“ zu schreiben. Hier ist ein Programm zum Schreiben von Gedichten, das PRINT und INPUT verwendet.

```
10 PRINT "WIE HEISSEN SIE?"
20 INPUT N$
30 PRINT "EIN GEDICHT VON"
40 PRINT N$
50 PRINT "NENNEN SIE EIN WORT, DAS"
60 PRINT "SICH AUF WISSEN REIMT"
70 INPUT A$
80 PRINT "HIER DAS GEDICHT:"
90 PRINT "VON COMPUTERN WOLLTE
        ICH NICHTS WISSEN"
100 PRINT "HEUTE MÖCHTE ICH SIE
          NICHT MEHR "
110 PRINT A$
120 END
```

Diese Zeile druckt Ihren Namen aus.

Diese Zeile druckt Ihr Wort aus.



```
RUN
WIE HEISSEN SIE?
? TOM
EIN GEDICHT VON TOM
NENNEN SIE EIN WORT, DAS
SICH AUF WISSEN REIMT
? MISSEN
HIER DAS GEDICHT:
VON COMPUTERN WOLLTE
ICH NICHTS WISSEN
HEUTE MÖCHTE ICH SIE
NICHT MEHR
MISSEN
```

Ihr Name und ein Wort

RUN drücken und noch einmal mit einem anderen Wort versuchen.

Das Programm bewirkt, daß der Computer die Antwort auf die Frage nach Ihrem Namen in N$ speichert und in Zeile 40 ausdruckt. Das von Ihnen gewählte Wort wird in A$ gespeichert und als Teil des Gedichts in Zeile 110 ausgedruckt. Wenn Sie einen Computer haben, lassen Sie das Programm mehrmals durchlaufen, jedesmal mit anderen Wort-Eingaben in Zeile 70.

Programmierübung

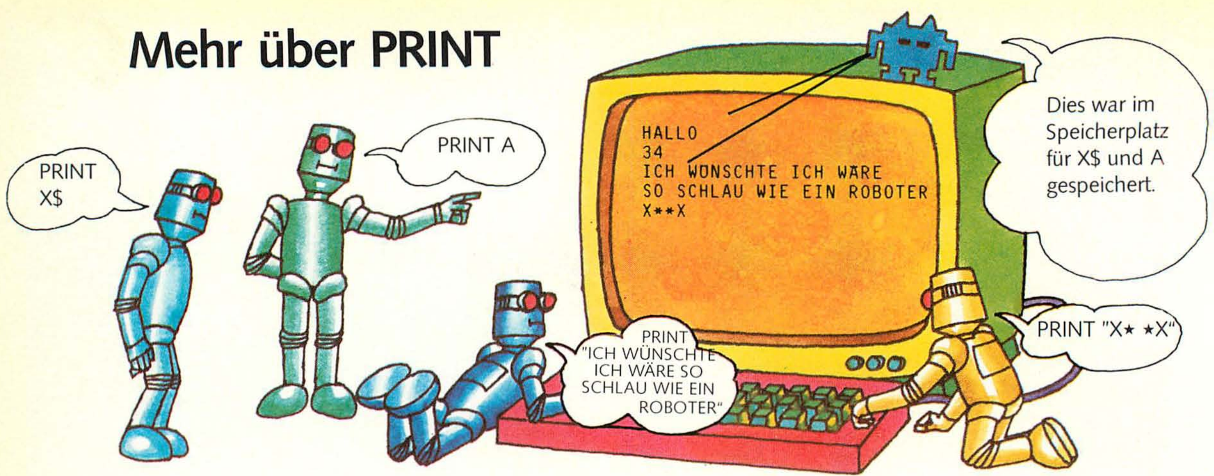
Versuchen Sie ein Programm zu schreiben, das nach Ihrem Namen fragt, dann „Guten Tag“ ausgibt, gefolgt von Ihrem Namen und einer Nachricht.

Eingabe-Regeln für Programme

1. Ehe man ein neues Programm eingibt, drückt man NEW. Dies bewirkt, daß alle alten Programme und Variablen aus dem Speicher des Computers verschwinden.
2. Beim Eingeben des Programms nicht vergessen, am Ende jeder Zeile RETURN zu drücken (oder das entsprechende Wort für Ihren Computer)!
3. Nachdem das Programm eingegeben ist, sollten alle Zeilen auf dem Bildschirm nochmals auf Tippfehler überprüft werden. Prüfen Sie auch, ob keine Zeile fehlt.
4. Dann CLS (oder das entsprechende Wort für Ihren Computer) eingeben, um das Programm vom Bildschirm zu entfernen. Um das Programm zu starten, RUN eingeben.
5. Um das Programm nochmals zu überprüfen und um eventuell eine Zeile zu ändern, LIST eingeben. Um nochmals eine bestimmte Zeile zu überprüfen, kann man normalerweise LIST zusammen mit der Zeilennummer eingeben; dies sollte man jedoch prüfen, da diese Anweisung von Computer zu Computer etwas verschieden sein kann.
6. Die Unterbrechung eines laufenden Programms erfolgt mit BREAK oder ESCAPE. Suchen Sie den passenden Ausdruck in Ihrem Computer-Handbuch. Bei manchen Computern bewirkt ESCAPE das Löschen des gesamten Programms! Das Programm kann wieder gestartet werden, indem man RUN eingibt.

Hinweise, wie man Fehler findet, stehen auf Seite 88/89.

Mehr über PRINT



Bisher haben wir gelernt, wie man PRINT zur Ausgabe von Wörtern und Zahlen auf dem Bildschirm und zur Ausgabe des Inhalts von Variablen verwendet. Im folgenden wird gezeigt, wie man Komma und Semikolon einsetzt, um Text auf dem Bildschirm anzuordnen. Man kann PRINT auch zur Ausführung von Rechnungen durch den Computer benutzen. Wie das gemacht wird, steht unten auf dieser Seite. Auf der nächsten Seite finden Sie weitere Einzelheiten zum Thema Variablen.

Komma und Semikolon

```
10 PRINT "HIER IST VIEL",
20 PRINT "PLATZ"
```

Komma

```
10 PRINT "HIER IST KEIN";
20 PRINT "PLATZ"
```

Semikolon

```
10 PRINT "HIER IST SEHR"
20 PRINT
30 PRINT "VIEL PLATZ"
```

```
HIER IST VIEL       PLATZ

HIER IST KEINPLATZ

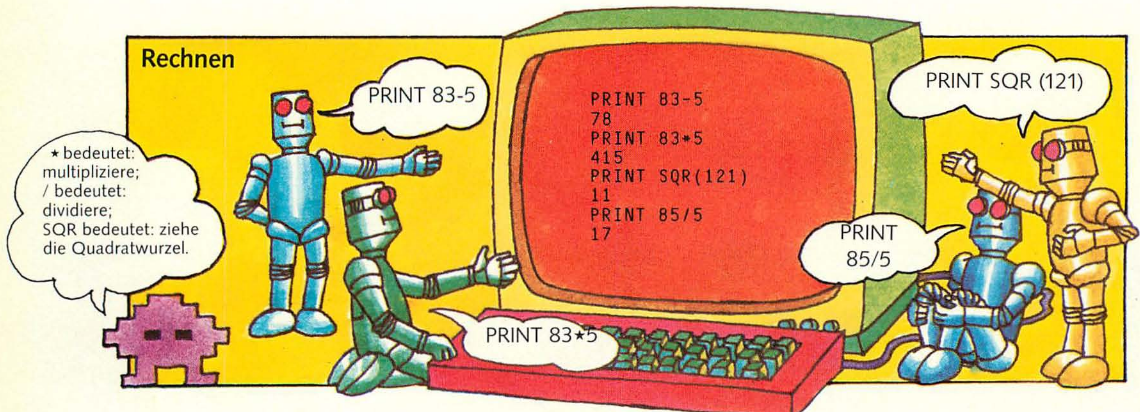
HIER IST SEHR

VIEL PLATZ
```

Das alleinstehende Wort PRINT bewirkt eine leere Zeile.

Diese Zeilen zeigen, wie man Komma und Semikolon verwendet, um dem Computer zu sagen, wo er den nächsten Buchstaben setzen soll. Ein Komma läßt ihn etwas vorrücken, bei einem Semikolon bleibt er, wo er ist.

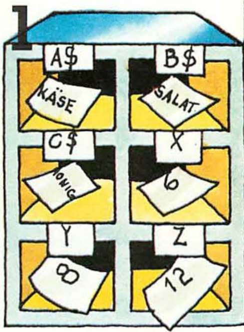
Das Bild oben zeigt die ausgegebenen Zeilen auf dem Bildschirm. Das alleinstehende Wort PRINT bewirkt eine leere Zeile.



So wird PRINT angewendet, um Rechnungen auszuführen. Man benutzt dazu die üblichen Zeichen für Addition und Subtraktion, * für Multiplikation und / für Division.

Der Computer kann auch viel kompliziertere Rechenarbeiten ausführen: Sinus, Cosinus, Quadratwurzel usw.

1



2

Bei den meisten Computern muß man auf beiden Seiten der Variablen einen Zwischenraum innerhalb der Anführungszeichen frei lassen.

Zwischenräume

```
PRINT "ICH HABE ";X;" WURST UND ";A$;" BRÖTCHEN GEGESSEN"
ICH HABE 6 WURST UND KÄSE BRÖTCHEN GEGESSEN

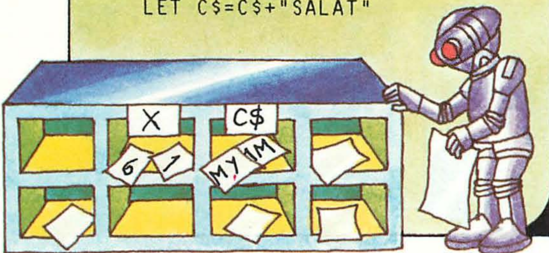
PRINT "ICH HABE ";Z;" WURST UND ";C$;" BRÖTCHEN GEGESSEN"
ICH HABE 12 WURST UND HONIG BRÖTCHEN GEGESSEN
```

Nur die Variablen selbst auszudrucken, ist nicht sehr sinnvoll. Besser ist es, sie in einen Text einzufügen. Um Wörter und Variablen zusammen auszudrucken, müssen die Wörter wie gewöhnlich in Anführungszeichen stehen, und die Variable muß auf jeder Seite ein Semikolon haben, wie oben gezeigt. Wenn man die Information räumlich besser aufteilen will, kann man Kommas anstelle der Semikolons einsetzen.

3

```
LET X=X+1

LET C$=C$+"SALAT"
```



Man kann in einem Programm den Inhalt der Speicherplätze verändern, wie im Bild gezeigt. Für den Computer bedeuten diese Anweisungen folgendes: Addiere 1 zu der Zahl im Speicherplatz X und füge „SALAT" zu den Buchstaben in C$ hinzu.

4

Zwischenräume

```
PRINT "ICH HABE ";X;" WURST UND ";
A$;" BRÖTCHEN GEGESSEN"

ICH HABE 7 WURST UND
KÄSE BRÖTCHEN GEGESSEN
```

Bei wiederholtem Ausdruck der Variablen erscheinen die neuen Zahlen und Wörter, die jetzt in den Speicherplätzen gespeichert sind.

5

```
10 LET A=9
20 LET B=7
30 PRINT A*B      ← Multipliziere
40 PRINT A/B      ← Dividiere
50 END
RUN
63
1.28571
```

Man kann Variablen auch multiplizieren oder dividieren, wie im Programm oben gezeigt wird. Der Computer sucht die Zahlen in den Speicherplätzen und berechnet die Ergebnisse.

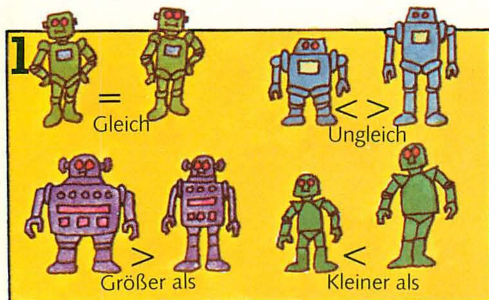
Programmierübungen

1. Schreiben Sie ein Programm, das Zahlen zu den Variablen des Programms von Bild 5 addiert (bzw. von den Variablen abzieht), so daß die Resultate 100 und 1 in einer Zeile mit einem Zwischenraum ausgedruckt werden.
2. Ändern Sie die Zeilen 30 und 40 so, daß die Zahlen, die Rechenoperation und das Resultat ausgedruckt werden, z. B. "9 mal 3 gleich 63".
3. Ändern Sie das Programm in der Übung auf Seite 15 so, daß Ihr Name und die Nachricht in einer Zeile ausgedruckt werden.

Wie der Computer Dinge vergleicht

Zu den nützlichsten Aufgaben, die ein Computer erledigen kann, gehört es, Informationen zu vergleichen und entsprechend den Ergebnissen zu handeln. Dazu verwendet man die Worte IF (wenn)... THEN (dann).





Der Computer kann mit Hilfe verschiedener Tests Informationen miteinander vergleichen. Die Symbole für diese Tests sind oben zu sehen. Er kann testen, ob zwei verschiedene Daten gleich sind, verschieden sind oder ob ein Datum größer oder kleiner als das andere ist.

2

```
IF A=B THEN PRINT "BEIDE SIND GLEICH"
IF A>B THEN PRINT "A IST GRÖSSER"
IF A<B THEN PRINT "A IST KLEINER"
IF A<>B THEN PRINT "SIE SIND UNGLEICH"
```

Diese Zeilen zeigen, wie die Symbole mit IF und THEN benutzt werden, um den Computer zu veranlassen, zwei Daten miteinander zu vergleichen. Man kann beliebige Arten von Daten vergleichen: Wörter, Zahlen und Variablen, d. h. auch die Inhalte von Speicherplätzen.

3 Wetter-Programm

```
10 PRINT "WIE IST DAS WETTER HEUTE?"
20 INPUT W$
30 IF W$="REGEN" THEN PRINT "REGENSCHIRM"
40 IF W$="SONNE" THEN PRINT "BADEHOSE"
50 END
```



4

```
RUN
WIE IST DAS WETTER HEUTE?
? SONNE
BADEHOSE
RUN
WIE IST DAS WETTER HEUTE?
? REGEN
REGENSCHIRM
```

Dies ist ein Programm mit IF und THEN. In Zeile 20 speichert der Computer das Wort, das als Variable W$ eingegeben wurde. In den Zeilen 30 und 40 prüft er, ob das Wort in W$ entweder "Regen" oder "Sonne" heißt. Wenn ja, gibt er eine der vorgegebenen Antworten aus. Wird in Zeile 20 ein anderes Wort eingegeben, dann geschieht nichts. Man kann jedoch die Wörter in den Zeilen 30 und 40 ändern und dann versuchen, eines der neuen Wörter einzugeben.

5 Alter-Programm

```
10 PRINT "WIE ALT BIST DU?"
20 INPUT A
30 IF A>16 THEN PRINT "OPA"
40 IF A<16 THEN PRINT "BABY"
50 IF A=16 THEN PRINT "PARTNER"
RUN
WIE ALT BIST DU?
? 16
PARTNER
```

6 Französisch-Lektion

```
10 PRINT "WIE HEISST ROT IN FRANZÖSISCH?"
20 INPUT A$
30 IF A$="ROUGE" THEN PRINT "RICHTIG"
40 IF A$<>"ROUGE" THEN PRINT "FALSCH, ROUGE"
RUN
WIE HEISST ROT IN FRANZÖSISCH?
? BLEU
FALSCH, ROUGE
```

Im Alter-Programm vergleicht der Computer die Eingabe A mit der Zahl 16. Wenn sie größer ist als 16, druckt er "Opa". Ist sie kleiner als 16, druckt er "Baby"; ist sie 16, druckt er "Partner". Im anderen Programm druckt der Computer eine von zwei möglichen Antworten aus, je nachdem, ob A$ gleich "rouge" ist oder nicht.



Programmierübung: Schreiben Sie ein Programm, in dem der Computer Ihnen Rechenaufgaben stellt und dann entweder "richtig" oder die richtige Antwort ausgibt.

Programmverzweigungen

1

```
IF A=6 THEN LET A$="SECHS"
IF X=Y-2 THEN LET Z=0
IF S=T THEN STOP
IF R<10 THEN GOTO 30
```

Dies bewirkt, daß der Computer zu Zeile 30 geht.

2

Diese beiden Zeilen bewirken einen Sprung zu anderen Programmzeilen.

```
10 PRINT K$
20 IF K$="JA" THEN GOTO 100
30 IF K$="NEIN" THEN GOTO 200
.
.
100 PRINT "DU HAST JA GETIPPT"
.
.
200 PRINT "DU HAST NEIN GETIPPT"
210 END
```

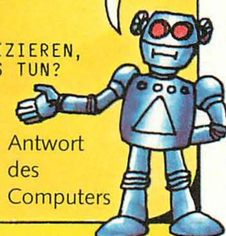
Im Anschluß an das Wort THEN kann man dem Computer fast jede Anweisung geben. Eine nützliche Anweisung besteht z. B. darin, in eine andere Zeile zu springen. (Bei den meisten Computern, nicht jedoch beim ZX81, kann man das Wort GOTO weglassen.) In Programmen mit GOTO braucht man normalerweise eine STOP-Anweisung, sonst wiederholt der Computer das Programm ständig.

Mathematik-Programm

```
10 PRINT "GIB EINE ZAHL EIN"
20 INPUT A
30 PRINT "GIB EINE WEITERE ZAHL EIN"
40 INPUT B
50 PRINT "MÖCHTEST DU ADDIEREN,"
60 PRINT "SUBTRAHIEREN, MULTIPLIZIEREN,"
65 PRINT "DIVIDIEREN ODER NICHTS TUN?"
70 INPUT C$
80 IF C$="ADDIEREN" THEN PRINT A+B
90 IF C$="SUBTRAHIEREN" THEN PRINT A-B
100 IF C$="MULTIPLIZIEREN" THEN PRINT A*B
110 IF C$="DIVIDIEREN" THEN PRINT A/B
120 IF C$="NICHTS TUN" THEN STOP
130 GOTO 10
```

```
RUN
GIB EINE ZAHL EIN
? 17
GIB EINE WEITERE ZAHL EIN
? 184
MÖCHTEST DU ADDIEREN,
SUBTRAHIEREN, MULTIPLIZIEREN,
DIVIDIEREN ODER NICHTS TUN?
? ADDIEREN
201
GIB EINE ZAHL EIN
?
```

DAS PROGRAMM ENDET ERST DANN, WENN SIE STOP EINGEBEN



In diesem Programm werden die von Ihnen eingegebenen Zahlen A und B und Ihre Anweisungen in C$ gespeichert. In Zeile 80 bis 120 vergleicht der Computer C$ mit fünf verschiedenen Wörtern und führt die Anweisung aus, sobald er das richtige Wort gefunden hat. Er übergeht dabei die Zeilen, die nicht relevant sind.

Alter-Rate-Programm

1

```
10 PRINT "RATE MEIN
ALTER"
20 INPUT R
30 IF R<> 14 THEN PRINT
   "RATE NOCH EINMAL"
40 IF R<> 14 THEN
GOTO 20
50 PRINT "RICHTIG"
60 END
```

2

```
RUN
RATE MEIN ALTER
? 15
RATE NOCH EINMAL
? 14
RICHTIG
```

3

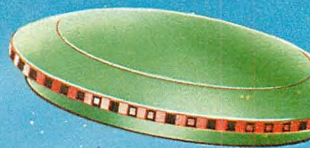
KÖNNEN SIE DAZU DAS PROGRAMM SCHREIBEN?

```
RATE MEIN ALTER
? 15
JÜNGER
? 13
ÄLTER
14
RICHTIG
```

Dieses Programm läuft so oft, bis R=14 kommt. Wenn R=14 ist, übergeht der Computer die Zeilen 30 und 40 und druckt "richtig".

Können Sie das Programm so abändern, daß es Ihnen einige Hinweise gibt, so wie in Bild 3 gezeigt?

Übungen mit BASIC



In den Programmen auf diesen beiden Seiten kommen die meisten der bisher behandelten BASIC-Ausdrücke vor. Das erste Programm ist ein Spiel für zwei Spieler mit dem Computer. Wenn Sie keinen Computer haben, studieren Sie die Programme und versuchen Sie zu verfolgen, wie sie funktionieren.

Raumkommando

```
10 PRINT "GEGNER: QUADRAT NACH RECHTS:"
20 INPUT A
30 PRINT "GEGNER: QUADRAT NACH OBEN:"
40 INPUT B
50 CLS
60 PRINT "KOMMANDO: QUADRAT NACH RECHTS:"
70 INPUT C
80 PRINT "KOMMANDO: QUADRAT NACH OBEN:"
90 INPUT D
100 CLS
110 LET X=SQR((A-C)*(A-C)+(B-D)*(B-D))
120 PRINT "DU BIST JETZT"
130 PRINT X;"RAUMMEILEN ENTFERNT"
140 IF X<1.5 THEN PRINT "GEGNER AUFGESPÜRT"
150 IF X<1.5 THEN STOP
155 PRINT "BITTE NEUE POSITION EINGEBEN"
160 GOTO 10
170 END
```

Die Zeilen 10 bis 40 speichern die Koordinaten des Gegners in A und B.

Zeile 50 entfernt die Koordinaten des Gegners vom Bildschirm.

Die Zeilen 60 bis 90 speichern die Koordinaten des Kommandos in C und D.

Diese Zeile berechnet die Entfernung der beiden voneinander und speichert das Resultat in X.

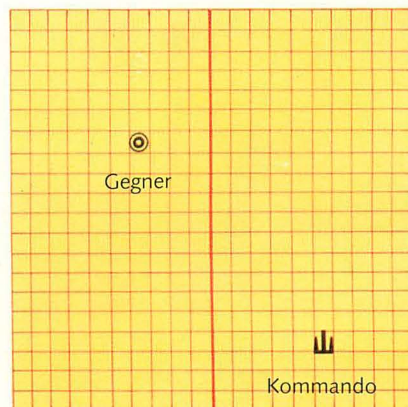
Ist X kleiner als 1,5, dann endet das Programm. Ist X größer als 1,5, dann läuft das Programm weiter.



In diesem Spiel ist der eine Spieler der Gegner und der andere das Raumkommando. Es versucht, den Gegner zu fangen. Jeder Spieler zeichnet für sich eine Karte und zeichnet dort seine Position ein. (Unten wird gezeigt, wie das gemacht wird.) Beide geben dem Computer die Gitterkoordinaten ihrer Position ein, und der Computer berechnet die Entfernung. Die Spieler benutzen die Ergebnisse des Computers, um die nächsten Schritte zu planen.

Spielplan

Um zu einer Karte zu kommen, zeichnet jeder Spieler ein Gitter mit 20 x 20 Quadraten, die numeriert werden (siehe Bild rechts). Der Gegner startet auf der linken Seite des Gitters, das Kommando startet auf der rechten Seite. Bei jedem Zug kann sich der Spieler je zwei Quadrate nach oben, unten, zur Seite oder diagonal bewegen; dann gibt er dem Computer die jeweilige neue Position ein. Sind die Spieler weniger als 1,5 Raumeinheiten (d. h. Quadrate) voneinander entfernt, dann hat das Kommando den Gegner gefangen.



Der schlaue Computer

In diesem Programm reagiert der Computer scheinbar klug auf die Beantwortung seiner Fragen. Wie das Programm läuft, steht unten auf dieser Seite. Das Programm verwendet INPUT in einer leicht veränderten Form, die es verkürzt und leichter lesbar macht.

1

```
10 INPUT "NENN MIR EINE ZAHL";N
20 INPUT "UND NOCH EINE";M
30 PRINT N;" MAL ";M;
" IST: ";N*M
```

Der BBC-Mikrocomputer braucht kein Semikolon

Bei den meisten Computern (Ausnahme: ZX81) kann man die INPUT-Zeile übersichtlicher gestalten, indem man Wörter in Anführungszeichen vor den Variablen-Namen stellt.

2

Beim ZX81 ist einzugeben:
10 PRINT „GIB MIR EINE ZAHL"
15 INPUT N

```
RUN
NENN MIR EINE ZAHL ? 10
UND NOCH EINE ? 8
10 MAL 8 IST 80
```



Wenn das Programm läuft, erscheint das Eingabe-Fragezeichen hinter dem Text.

Das Programm

```
5 LET C=0
10 PRINT "ICH MÖCHTE MICH MIT DIR UNTERHALTEN"
20 INPUT "ERZÄHLE MIR WAS DIR IN DIESER WOCHE
         PASSIERT IST";A$
30 READ B$
40 PRINT B$
50 INPUT C$
60 LET C=C+1
70 IF C=6 THEN GOTO 100
80 GOTO 30
90 DATA WARUM,WARUM DAS
95 DATA WIESO NUR,WARUM GENAU
98 DATA SAG MIR NUR WARUM, AUS WELCHEM GRUND
100 PRINT "ALSO LIEGT DER GRUND FÜR DEINE EINGABE"
110 PRINT "     ";A$
120 PRINT "IN WIRKLICHKEIT IN DEINER ANTWORT"
130 PRINT "     ";C$
140 PRINT "WIE SELTSAM!"
150 PRINT "LASS MICH NOCHMALS LAUFEN ZUR
          WEITEREN ERHELLUNG"
160 END
```

Dies ist die neue INPUT-Form. Ihre Antwort wird in A$ gespeichert.

In Zeile 30 sucht der Computer nach der ersten DATA-Zeile, nimmt das erste Textstück und speichert es in B$.

Dies läßt den Computer in derselben Zeile bleiben. (Zeile 40)

Ihre Antwort wird in C$ gespeichert. (Zeile 50)

Die Variable C in Zeile 60 und 70 ist ein Zähler, der zählt, wie oft das Programm wiederholt wird. Ist C = 6, sind alle DATA-Eingaben verbraucht, und der Computer springt zu Zeile 100.

Zeile 80 bewirkt Rücksprung zu Zeile 30, dabei werden die Daten in B$ durch den nächstfolgenden Text in der Datenliste ersetzt.

Die Leerstellen in Zeile 110 und 130 lassen freien Platz auf dem Bildschirm vor Ihren Antworten. Sie können beliebig viele Stellen freilassen.

Programmablauf

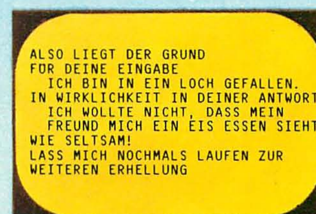






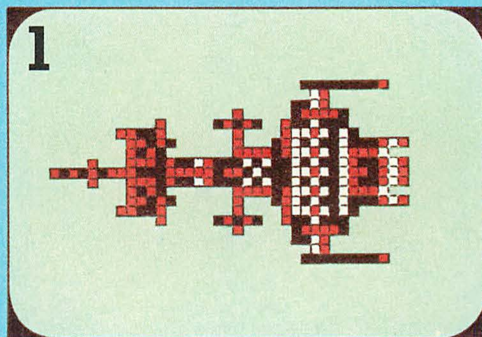




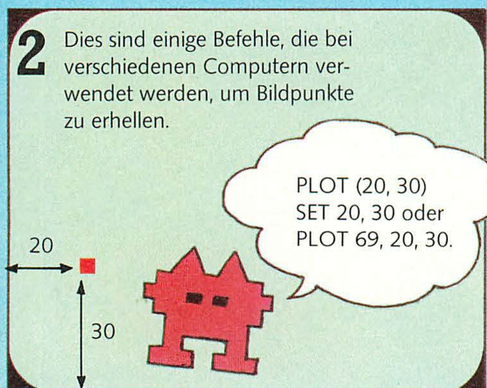


Computer-Grafik

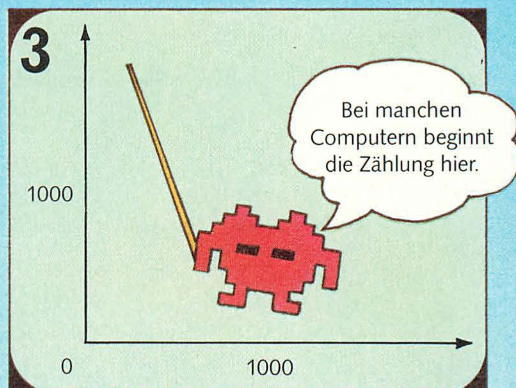
Der Computer kann Bilder „malen", indem kleine Flächen auf dem Bildschirm aufgehellt werden. Diese kleinen Flächen heißen *Bildpunkte*. Der Computer gibt jedem Bildpunkt eine entsprechende Anweisung, um ihn aufzuhellen. Die meisten Computer können die Bildpunkte auch mit verschiedenen Farben einfärben. Auf diesen beiden Seiten wird gezeigt, wie man BASIC anwendet, um einfache Bilder auf den Bildschirm zu bringen. Die hier gegebenen Anweisungen beziehen sich nur auf einfarbige Bilder.



Normalerweise kann man die Bildpunkte eines Computerbildes sehen. Computer mit großem Speicher können jedoch Bilder malen, die aus Tausenden winziger Bildpunkte bestehen. Diese Bilder heißen *hochauflösende Grafik*.



Diese Anweisung zur Aufhellung eines Bildpunktes ist von Computer zu Computer verschieden; normalerweise lautet sie PLOT (X, Y) oder ähnlich. X und Y sind die Koordinaten des Bildpunktes, d. h. X ist die Anzahl der Punkte nach rechts, und Y ist die Anzahl der Punkte nach oben.

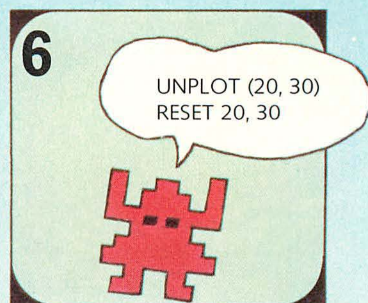


Bei einem Computer mit hochauflösender Grafik kann man 1000 Punkte in der Breite und 1000 Punkte in der Höhe des Bildschirms zeichnen. Ein weniger leistungsfähiger Computer hat ungefähr 60 x 40 Bildpunkte. (Prüfen Sie die Angaben für Ihren Computer, damit Sie beim Zeichnen nicht außerhalb des Bildschirms geraten.)





Computerbilder nennt man in der Regel *Grafiken*. Manche Computer benötigen einen speziellen Befehl, bevor man mit den Grafiken beginnen kann. So muß man beim BBC-Mikrocomputer z. B. das Wort MODE zusammen mit einer Zahl eingeben.*



Befehle wie UNPLOT (X, Y) schalten einen Bildpunkt aus. In den Programmen dieses Buches werden PLOT und UNPLOT verwendet. Wenn Sie einen Computer haben, prüfen Sie diese Befehle im Handbuch nach.

* Für die Programme in diesem Buch verwenden Sie MODE 5 auf dem BBC. Grafikbefehl: PLOT 69, X, Y, Löschbefehl: PLOT 71, X, Y.

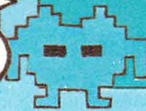
Grafik-Programm



1

```
10 PRINT "GIB 2
   ZAHLEN EIN"
20 INPUT X
30 INPUT Y
40 PLOT (X,Y)
50 GOTO 10
```

Drücken Sie NEWLINE oder RETURN nach jeder Zahleneingabe.



2

```
RUN
GIB 2 ZAHLEN EIN
? 24
? 24 ■
GIB 2 ZAHLEN EIN
? 30
? 15      ■
```

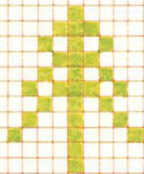
1. Bildpunkt

2. Bildpunkt

Dieses kurze Programm fragt nach zwei Zahlen und zeichnet dann den Bildpunkt mit diesen Zahlen als Koordinaten. Achten Sie beim Ausprobieren dieses Programms darauf, daß die eingegebenen Zahlen innerhalb des Zahlenbereichs Ihres Bildschirms liegen.

Zeile 50 bewirkt, daß das Programm ständig wiederholt wird; die einzige Möglichkeit, es zu beenden, besteht mit dem Befehl BREAK (oder dem entsprechenden Wort für Ihren Computer). Können Sie einen Zähler anbringen, der das Programm z. B. sechsmal laufen läßt (siehe Seite 67)?

So malt man ein Bild mit dem Computer

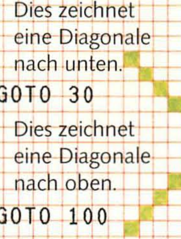


```
10 LET X=10
20 LET Y=10
30 PLOT(X,Y)
40 LET X=X+1
50 LET Y=Y-1
60 IF X<14 THEN GOTO 30

100 LET Y=Y+1
110 LET X=X+1
120 PLOT(X,Y)
130IF X<20 THEN GOTO 100
```

Dies zeichnet eine Diagonale nach unten. (Zeilen 30–50)

Dies zeichnet eine Diagonale nach oben. (Zeilen 100–120)





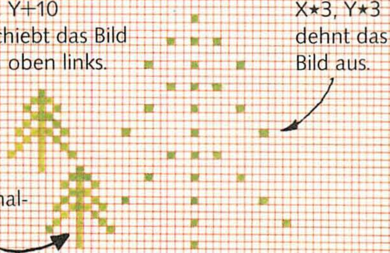
1 zu X hinzugezählt (und nicht zu Y) ergibt eine waagerechte Linie.



1 zu Y hinzugezählt (und nicht zu X) ergibt eine senkrechte Linie.

Zeichnen Sie das Bild zuerst auf kariertes Papier und berechnen Sie dann die Koordinaten der Quadrate.

Arbeiten Sie anschließend das vollständige Programm zum Zeichnen aller Quadrate aus. Geben Sie für X und Y Ausgangswerte an, addieren oder subtrahieren Sie diese und wiederholen Sie Teile des Programms. Damit erreichen Sie, daß der Computer Bildpunkte so zeichnet, wie es oben gezeigt wird.



Wenn das Programm einmal geschrieben ist, ist es leicht, das Bild durch Veränderung der Zahlen abzuändern. Man kann es auf einen anderen Platz auf dem Bildschirm verschieben, indem man die Anfangswerte ändert, oder man kann es vergrößern, indem man alle Zahlen mit 3 multipliziert.

Eine andere Möglichkeit

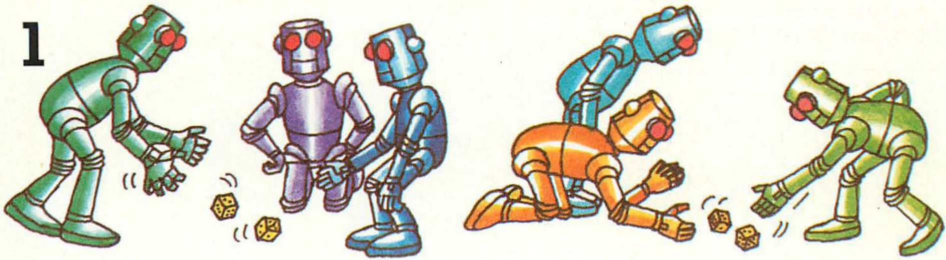


PLOT ermöglicht nur sehr einfache Bilder. Für kompliziertere Bilder braucht man spezielle Geräte, z. B. ein Grafiktablett. Auf das Tablett wird eine Zeichnung gemalt, die dann mit einem „Puck" abgetastet wird. Auf diese Weise werden automatisch die Koordinaten in den Computer übertragen.



Programmierübung: Schreiben Sie ein Programm, das die Anfangsbuchstaben Ihres Namens auf den Bildschirm malt. Ein Beispiel dafür steht auf Seite 90.

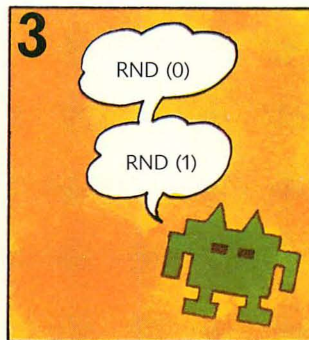
Computer-Spiele



Beim Würfeln kann man die Ergebnisse nicht vorhersagen. Alle Zahlen zwischen 1 und 6 haben die gleiche Chance. Man kann unvorhersehbare Zahlen auch mit einem Computer herstellen. Solche Zahlen heißen *Zufallszahlen.*

Im Computer ist ein besonderes Programm für Zufallszahlen gespeichert. Manchmal wird eine bestimmte Zahl mehrere Male wiederholt. In den Folgen vieler Zufallszahlen erscheinen jedoch alle Zahlen ungefähr gleich oft.

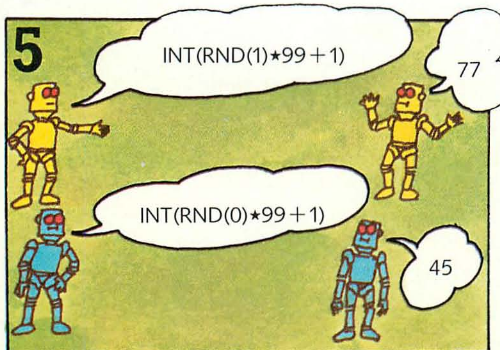


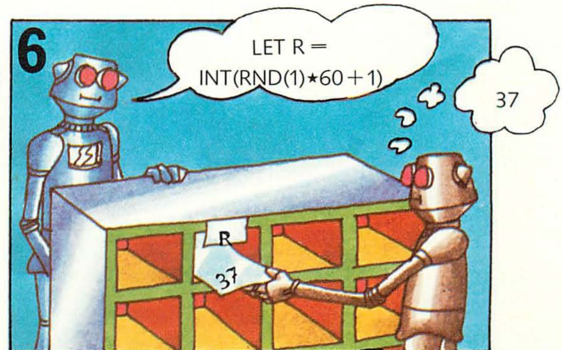




Wenn man RND eingibt, gibt der Computer eine Zufallszahl aus. Bei manchen Computern muß man im Anschluß an RND eine 1 oder 0 in Klammern eingeben. Wenn Sie einen Computer haben, überprüfen Sie das in Ihrem Handbuch.

Der RND-Befehl bewirkt die Angabe einer Zahl unter 1. Bei manchen Computern kann man RND und eine Zahl in Klammern angeben, z. B. RND (99). Dann wird eine ganze Zahl zwischen 1 und der Zahl in Klammern ausgegeben.





Bei manchen Computern muß man das Wort INT (Kurzform für engl. integer = ganze Zahl) eingeben, gefolgt von RND – entweder RND (1) oder RND (0). Dann multipliziert man mit der größten Zahl, die man haben möchte, und addiert 1. Auf diese Weise erhält man eine Zahl, die größer ist als 1.

Diese Anweisung bedeutet folgendes: Wähle eine Zufallszahl aus und speichere sie in der Variablen R. In diesem Buch bedeutet z. B. INT(RND(1)*60+1): Wähle eine Zufallszahl zwischen 1 und 60. Eventuell ist diese Anweisung für Ihren Computer (entsprechend den Angaben im Handbuch) etwas abzuändern.



Programmierübung: Schreiben Sie ein Programm, das den Computer veranlaßt, eine Zufallszahl zwischen 10 und 20 auszugeben.

Angriff im Weltraum



Dies ist ein Spielprogramm, in dem Zufallszahlen verwendet werden. Das Spiel geht so: Ihr Raumschiff wird von gegnerischen Fahrzeugen angegriffen. Der Computer des Raumschiffs ortet die Maschinen und gibt ihre codierten Positionen aus. Um die Fahrzeuge vom Bildschirm abzuräumen, muß man die Entfernung zu ihnen berechnen, indem man die Codes multipliziert und dann die Anwort eingibt.

```
10 LET C=0
20 LET A=INT(RND(1)*20+1)
30 LET B=INT(RND(1)*20+1)
40 PRINT "CODES DES GEGNERS"
45 PRINT "SIND ";A,B;" FEUER"
50 INPUT X
60 LET C=C+1
70 IF X=A*B THEN PRINT "GEGNER
   ABGESCHOSSEN"
80 IF X<>A*B THEN PRINT "ZIEL VERFEHLT"
90 IF C<6 THEN GOTO 20
100 END
```

C ist ein Zähler, der zählt, wie oft das Programm läuft. In Zeile 60 wird jedesmal 1 zu C addiert. (Zeile 10)

In diesen zwei Zeilen entstehen Zufallszahlen für die Codes des gegnerischen Fahrzeugs. Diese werden in A und B gespeichert. (Zeilen 20, 30)

Ihre Zahl wird in X gespeichert. (Zeile 50)

In Zeile 70 und 80 prüft der Computer, ob die Antwort richtig ist.

Diese Zeile bewirkt die Wiederholung des Programms, wenn C kleiner ist als 6. (Zeile 90)

So läuft das Programm

Das rechte Bild zeigt den Ablauf des Programms. Wenn die richtige Antwort – Multiplikation der beiden Zahlen – eingegeben wird, dann gibt der Computer "Gegner abgeschossen" aus. Weicht die Antwort von AxB ab, dann gibt der Computer "Ziel verfehlt" aus.

```
RUN
CODES DES GEGNERS
SIND 17        3 FEUER
? 41
ZIEL VERFEHLT
CODES DES GEGNERS
SIND 11        5 FEUER
? 55
GEGNER ABGESCHOSSEN
CODES DES GEGNERS
SIND 13        6 FEUER
```

Das Komma in Zeile 45 bewirkt den Abstand der Zahlen voneinander.

Programmierübung

Versuchen Sie, einen weiteren Zähler ins Programm einzubauen, der die Zahl der Treffer bestimmt und die Trefferpunkte am Ende des Spiels angibt. Dazu muß man eine Variable S mit dem Startwert 0 festlegen und bei jedem Treffer 1 addieren.

Programm für Zufallsmuster

```
5 CLS
10 LET X=INT(RND(1)*30+1)
20 LET Y=INT(RND(1)*30+1)
30 PLOT (X,Y)
40 GOTO 10
```

Dadurch wird der Bildschirm freigemacht, ehe die Bildpunkte auf dem Bildschirm erscheinen. (Zeile 5)

Die Zufallszahlen müssen innerhalb des Bildschirms liegen. (Zeilen 10, 20)

Diese Zeile bewirkt die endlose Wiederholung des Programms. (Zeile 40)

Die Computer-Befehle für CLS, RND und PLOT können je nach Computer verschieden sein.



In diesem Programm werden Zufallszahlen verwendet, um Bildpunkte auf den Bildschirm zu zeichnen. In den Zeilen 10 und 20 werden Zufallszahlen zwischen 1 und 30 erzeugt und in X und Y gespeichert. Zeile 30 bewirkt das Zeichnen der Bildpunkte mit den Koordinaten X, Y. Während sich der Bildschirm mit Bildpunkten füllt, erscheinen immer weniger neue Punkte, weil viele bereits vorhanden sind. Um das Programm zu stoppen, gibt man BREAK oder ESCAPE ein bzw. das für Ihren Computer passende Wort.

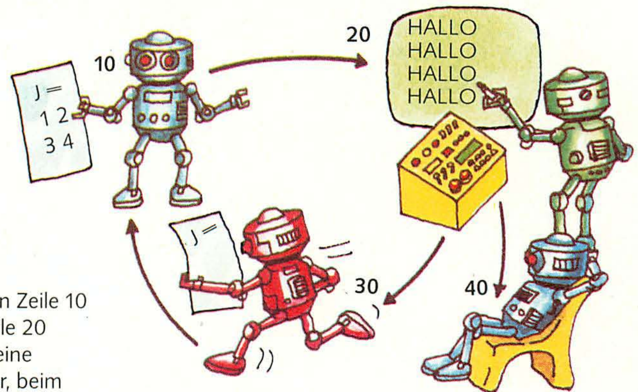
Schleifen

Oft ist es in einem Programm erforderlich, daß der Computer einen bestimmten Vorgang mehrere Male wiederholt. Auf Seite 67 haben wir gesehen, wie solche Wiederholungen von Programmteilen mit GOTO und einer Variablen erfolgen, die als Zähler dient. Eine andere Möglichkeit besteht darin, dieselbe Zeile mehrmals zu wiederholen, indem die Befehle FOR...TO und NEXT verwendet werden. Dies nennt man *Schleife*.

1 Hallo-Schleife

```
10 FOR J=1 TO 6
20 PRINT "HALLO"
30 NEXT J
40 END
```

Schleife



Dieses Programm enthält eine Schleife von Zeile 10 bis 30. Sie bewirkt, daß der Computer Zeile 20 sechsmal wiederholt. Der Buchstabe J ist eine Variable. Zeile 10 veranlaßt den Computer, beim ersten Durchlauf J = 1 zu setzen, beim zweiten Durchlauf J = 2, dann 3 usw. bis 6. Zeile 20 bewirkt den Ausdruck des Wortes "Hallo". Zeile 30 läßt den Computer zurückgehen, um den nächsten Wert für J zu suchen. Sobald J = 6, springt der Computer zu Zeile 40.

2 Programm „Dumme Summe"

```
10 FOR J=1 TO 8
20 PRINT "2 PLUS 2 IST 5"
30 NEXT J
40 PRINT
50 PRINT "NUR EIN SCHERZ!"
60 END
```

Schleife

Bei manchen Computern gibt es kein Ausrufezeichen, in diesem Fall können Sie es auslassen.

```
2 PLUS 2 IST 5
2 PLUS 2 IST 5
2 PLUS 2 IST 5
2 PLUS 2 IST 5
2 PLUS 2 IST 5
2 PLUS 2 IST 5
2 PLUS 2 IST 5
2 PLUS 2 IST 5

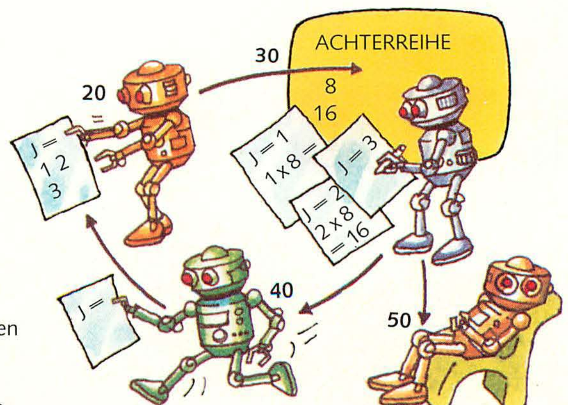
NUR EIN SCHERZ!
```

In diesem Programm bewirkt die Schleife von Zeile 10 bis 30, daß der Computer Zeile 20 achtmal wiederholt. Jedesmal, wenn Zeile 20 an der Reihe ist, gibt der Computer die gleiche „dumme Summe" aus. Nachdem dies achtmal durchgeführt wurde, fährt der Computer mit dem restlichen Programm fort. Zeile 40 bewirkt lediglich eine leere Zeile.

3 Achterreihen-Programm

```
10 PRINT "ACHTERREIHE"
20 FOR J= 1 TO 12
30 PRINT J*8
40 NEXT J
50 END
```

Schleife



In diesem Programm wird J sowohl als Schleifenzähler als auch als Teil der Multiplikation J*8 verwendet. Zeile 20 gibt die Anweisung, zuerst J = 1, dann 2, 3 usw. bis 12 zu setzen. Zeile 30 nimmt den gerade bestehenden Wert von J, multipliziert ihn mit 8 und gibt die Antwort aus. Zeile 40 bewirkt einen Rücksprung auf Zeile 20 zur Suche des nächsten Wertes für J.

So macht man Muster mit dem Computer

FOR... NEXT-Schleifen sind nützlich, wenn man Muster herstellen will, die aus der Wiederholung einer einfachen Figur bestehen. Das Programm für dieses Muster ist zu lang, als daß es hier vollständig abgedruckt werden könnte, aber es sieht ungefähr so aus:

```
10 FOR I = 1 TO 45
20 Zeichne ein Rechteck und verändere
   jedesmal ein wenig seine Position.
30 NEXT I
40 END
```

Schritte

Manchmal ist es nützlich, den Wert von J in anderen als Einer-Schritten zu ändern. So möchte man z. B. in Dreier-Schritten nach oben oder lieber in Siebener-Schritten nach unten gehen. Dazu dient das Wort STEP (engl. step = Schritt). Im folgenden Programm bewirkt STEP –1, daß jedesmal, wenn der Computer die Schleife in Zeile 10 bis 40 abarbeitet, J um 1 kleiner wird.

Programm "Gefräßiger Computer"

Die Zahl 2 stoppt die Schleife nach J = 2, d. h. wenn noch ein Keks da ist.

```
5 CLS
10 FOR J=7 TO 2 STEP -1
20 PRINT "ES SIND NOCH ";J;" KEKSE DA"
30 NEXT J
40 PRINT
50 PRINT "ICH PLATZE"
60 FOR K=1 TO 1000
70 REM VERZÖGERUNG
80 NEXT K
90 PRINT
100 PRINT "*W*U*M*M*"
```

Schleife

Schleife

```
ES SIND NOCH 7 KEKSE DA
ES SIND NOCH 6 KEKSE DA
ES SIND NOCH 5 KEKSE DA
ES SIND NOCH 4 KEKSE DA
ES SIND NOCH 3 KEKSE DA
ES SIND NOCH 2 KEKSE DA

ICH PLATZE

*W*U*M*M*
```

Manche Computer arbeiten etwas langsamer; in diesem Fall gibt man in Zeile 60 eine niedrigere Zahl ein, z. B. 500 oder 250.



In diesem Programm sind zwei Schleifen. Die Schleife von Zeile 10 bis 30 bewirkt, daß Zeile 20 sechsmal ausgedruckt wird. Jedesmal wird der Wert von J um 1 vermindert, und der entsprechende Wert von J wird in Zeile 20 ausgegeben. Bei der Schleife von Zeile 60 bis 80 tut der Computer nichts. Er durchläuft lediglich alle Werte für K von 1 bis 1000, dadurch entsteht eine kleine Pause. Zeilen, die mit REM (Kurzform für engl. remark = Bemerkung) beginnen, ignoriert der Computer. REM-Zeilen dienen dazu festzuhalten, was das Programm macht.

Programmierübungen

1. Ändern Sie das Programm "Achterreihe" von Seite 26 so, daß es "1x8=" gefolgt vom Ergebnis ausdruckt.
2. Versuchen Sie ein Programm zu schreiben für das „1 mal N", d. h. ein Programm, das für jede beliebige Zahl N, die eingegeben wird, die entsprechende 1x1-Reihe ausgibt. Hinweis: Zunächst läßt man den Computer nach einer Zahl N fragen. Dann benutzt man eine Schleife, um die 1x1-Reihe zu berechnen und die Ergebnisse auszugeben. Am Ende des Programms könnte man noch einige Zeilen hinzufügen, die abfragen, ob die 1x1-Reihe für eine andere Zahl gewünscht wird; dann kann man das Programm erneut laufen lassen.

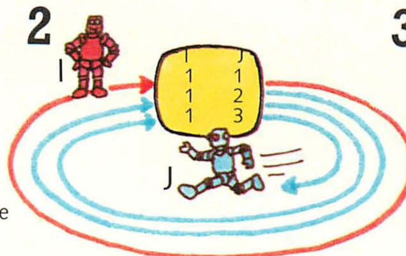
Tricks mit Schleifen

Hier finden Sie weitere Programme, in denen Schleifen vorkommen. Sie erfahren dabei, wie Schleifen innerhalb von Schleifen verwendet werden, um verschiedene Dinge gleichzeitig zu wiederholen. Solche Schleifen heißen *geschachtelte Schleifen.*

1 Geschachtelte Schleifen

```
5 PRINT "I","J"
10 FOR I=1 TO 3
20 FOR J=1 TO 3
30 PRINT I,J
40 NEXT J      ← J-Schleife
50 NEXT I      ← I-Schleife
60 END
```

2



3

I	J
1	1
1	2
1	3
2	1
2	2
2	3
3	1
3	2
3	3

Dieses Programm hat eine I-Schleife und eine J-Schleife. Die J-Schleife befindet sich innerhalb der I-Schleife. Jedesmal, wenn die I-Schleife bearbeitet wird, wird die J-Schleife dreimal wiederholt, wobei der Wert von J jedesmal ausgegeben wird. Das Bild oben zeigt das Ergebnis dieses Programms. Die Kommas bewirken, daß die Zahlen übersichtlich angeordnet sind.

Computer-Uhr

```
5 CLS
10 LET M=0
20 LET S=0
30 FOR M=0 TO 59
40 FOR S=0 TO 59
50 PRINT M;":";S
60 CLS
70 NEXT S      ← Sekunden-Schleife
80 NEXT M      ← Minuten-Schleife
90 END
```

0:45

Benutzen Sie diese „Verzögerungsschleife" zur richtigen Zeitanzeige:
54 FOR Z = 1 TO 100
58 NEXT Z



Fehler in Schleifen

```
10 FOR I=1 TO 4
20 FOR J=1 TO 4
30 PRINT I
40 PRINT J
50 NEXT I
60 NEXT J
```

Beide Teile einer geschachtelten Schleife müssen innerhalb der anderen Schleife sein.



Im Inneren eines Computers befindet sich eine elektronische „Uhr", die den Rhythmus für die gesamte Arbeit des Computers bestimmt. Die Uhr „tickt" zwischen ein- und viermillionenmal in der Sekunde. Das Programm oben bewirkt, daß der Computer sich wie eine Digitaluhr verhält. Das Programm enthält geschachtelte Schleifen: eine zählt die Sekunden, eine andere zählt die Minuten.

Die Sekundenschleife wird innerhalb jeder Minutenschleife 60mal ausgeführt. Wenn man dieses Programm auf einem Computer ausprobiert, wird es zunächst zu schnell ablaufen. Daher ist es ratsam, eine spezielle „Verzögerungsschleife" einzubauen. Damit erreicht man, daß die Computeruhr genauso schnell „tickt" wie eine richtige Uhr.

Zufallszahlen-Test

```
10 FOR I=1 TO 1000
20 LET R=INT(RND(1)*6+1)
30 IF R=1 THEN LET A=A+1
40 IF R=2 THEN LET B=B+1
50 IF R=3 THEN LET C=C+1
60 IF R=4 THEN LET D=D+1
70 IF R=5 THEN LET E=E+1
80 IF R=6 THEN LET F=F+1
90 NEXT I
100 PRINT "ENDE"
110 PRINT A,B,C
120 PRINT D,E,F
130 END
```

```
RUN
ENDE
162         168         167
160         187         156
```

Dieses Programm dauert sehr, sehr lange. Man kann es verkürzen, indem man die Zahl in Zeile 10 ändert: z. B. in 500 oder 250.

Dieses Programm zeigt an, ob RND wirklich funktioniert. Die Schleife von Zeile 10 bis 90 bewirkt, daß der Computer tausendmal eine Zufallszahl zwischen 1 und 6 auswählt. Dabei wird gezählt, wie oft jede Zahl in den Variablen A bis F genannt wird. Dann werden die Ergebnisse ausgedruckt.*

* Bei manchen Computern, z. B. ZX81 und BBC, sind zu Beginn des Programms zusätzliche Zeilen erforderlich, um jede Variable auf 0 zu setzen.

Programm "Muster wiederholen"

Dieses Programm verwendet geschachtelte Schleifen, um ein kleines Muster auf dem Bildschirm zu wiederholen. Das Programm sieht ziemlich kompliziert aus; wenn man es jedoch sorgfältig durchliest und überlegt, was in jeder Zeile geschieht, erkennt man rasch, wie es funktioniert. Die Form des Musters wird durch Zufallszahlen bestimmt, sie verändert sich daher bei jedem Programmdurchlauf.

Bei Computern mit hochauflösender Grafik kann man höhere Zufallszahlen verwenden.



```
5 CLS
10 LET A=INT(RND(1)*6+1)
20 LET B=INT(RND(1)*7+1)
30 LET C=INT(RND(1)*6+1)
40 LET D=INT(RND(1)*4+1)
50 INPUT "ZAHL DER PUNKTE IN
   BILDSCHIRMBREITE";BB
60 INPUT "ZAHL DER PUNKTE IN
   BILDSCHIRMHÖHE";BH
65 CLS
70 FOR I=0 TO BH STEP BH/6
80 FOR J=0 TO BB STEP BB/6
90 PLOT (J+A,I+B)
100 PLOT (J+A,I+C)
110 PLOT (J+C,I+D)
120 PLOT (J+B,I+D)
130 NEXT J
140 NEXT I
150 END
```

J-Schleife

I-Schleife

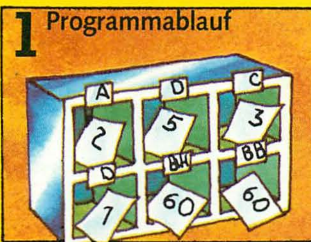
Diese Zeilen bestimmen die Zufallszahlen für das Muster und speichern sie in A, B, C und D.

Zeile 50 und 60 erfragen Breite (BB) und Höhe (BH) Ihres Bildschirms.

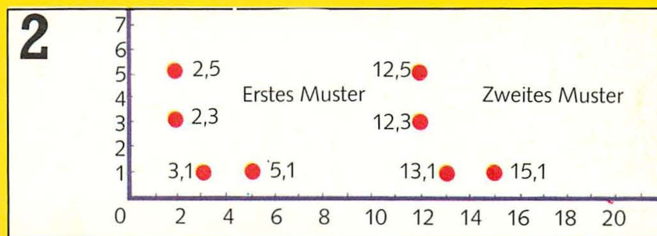
Die I-Schleife zählt, wie oft das Muster auf dem Bildschirm nach oben erscheint. I wächst jedesmal um die Höhe des Bildschirms (BH) geteilt durch 6; das Muster erscheint daher sechsmal in der Höhe des Bildschirms.

Bei jeder Wiederholung der Schleifen veranlassen die Zeilen 90 bis 120 den Computer, vier Bildpunkte zu zeichnen mit den gerade geltenden Werten für I und J plus Zufallszahlen.

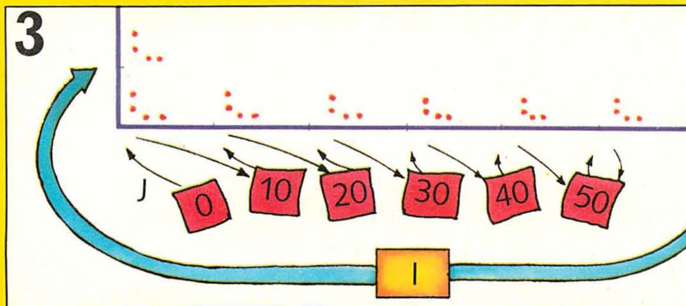
Die J-Schleife zählt, wie oft das Muster auf dem Bildschirm nach rechts erscheint. Diese Schleife arbeitet genauso wie die I-Schleife.



Wir nehmen einmal an, der Computer hat die Zufallszahlen 2, 5, 3 und 1 gewählt, und Breite und Höhe des Bildschirms sind jeweils 60.



Beim ersten Programmdurchlauf sind I und J gleich 0; der Computer benutzt also nur die Zufallszahlen, um das erste Punktmuster zu zeichnen. In Zeile 130 erfolgt ein Rücksprung, um den nächsten Wert für J zu bestimmen: J+60/6, d. h. 10. Dann wird das zweite Muster gezeichnet, und zwar mit J gleich Zufallszahlen plus 10. Auf diese Weise wiederholt sich das Muster in der Breite des Bildschirms.



Wenn dieses Programm auf Ihrem Computer nicht läuft, versuchen Sie es mit kleineren Zahlen für BH und BB.



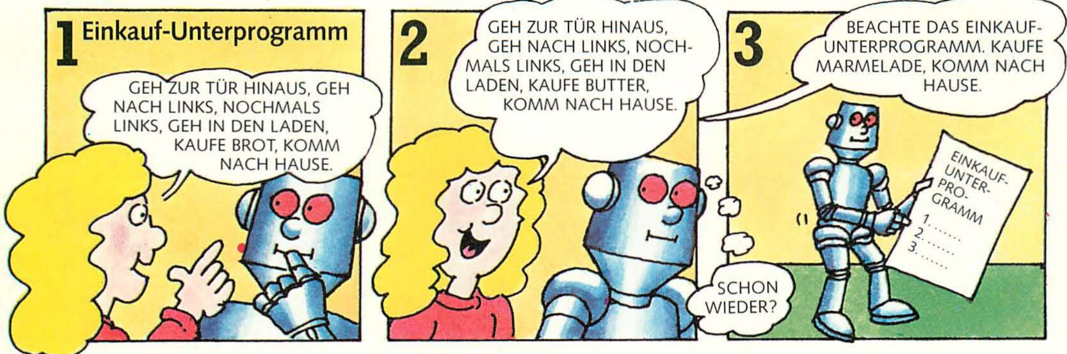
Der Computer wiederholt die J-Schleife sechsmal, er addiert jedesmal 10 zu J hinzu und zeichnet so das Muster in der Breite des Bildschirms. Dann erfolgt ein Rücksprung, um den Wert für I zu suchen; dieser ist gleich 10. J wird wieder gleich 0 gesetzt, und der Computer zeichnet die nächste Muster-Zeile mit I gleich 10 und J, das jedesmal um 10 wächst.



Programmierübung: Versuchen Sie ein Wiederholungsprogramm zu schreiben, das ein Raumfahrer-Muster auf dem Bildschirm erscheinen läßt. Auf Seite 91 finden Sie einige Hinweise dazu.

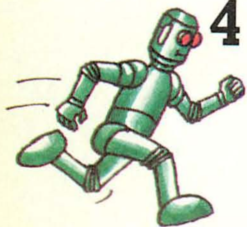
Unterprogramme

Ein Unterprogramm (engl. subroutine) ist eine Art Mini-Programm innerhalb eines Programms. Man führt damit eine spezielle Aufgabe aus, z. B. die Addition von Zahlen, Festhalten von Punktzahlen usw. Der Computer kann jederzeit veranlaßt werden, ein solches Unterprogramm auszuführen. Man erspart sich dadurch, diesen Programmteil jedesmal Zeile für Zeile auszuschreiben. Das gesamte Programm wird dadurch kürzer und leichter lesbar, und man kann es schneller in den Computer eingeben.



Angenommen, Sie hätten einen Roboter, den Sie so programmieren könnten, daß er Besorgungen erledigt. Wenn er einkaufen soll, müßten Sie ihm genau sagen, wie er zum Laden kommt.

Jedesmal wenn der Roboter etwas kaufen soll, müßten Sie ihm die gleichen Anweisungen geben. Es wäre viel einfacher, dem Roboter ein Einkauf-Unterprogramm zu geben und ihn zu veranlassen, sich gegebenenfalls darauf zu beziehen.



4 Einkauf-Programm

```
10 PRINT "WAS WILLST DU EINKAUFEN?"
20 INPUT X$
30 GOSUB 100
40 PRINT "NOCH ETWAS?"
50 INPUT M$
60 IF M$="JA" THEN GOTO 10
70 STOP

100 REM UNTERPROGRAMM EINKAUF
110 PRINT "GEH HINAUS, NACH LINKS"
120 PRINT "NOCHMALS LINKS, GEH IN DEN LADEN"
130 PRINT "KAUFE ";X$;" KOMM NACH HAUSE"
140 RETURN
```

Zeile 30 schickt den Computer zur ersten Zeile des Unterprogramms.

Das Wort STOP am Ende des Hauptprogramms ist erforderlich, damit der Computer nicht in das Unterprogramm hineingerät.

Es ist nützlich, ein Unterprogramm mit einer REM-Zeile zu kennzeichnen, damit man es leichter wiedererkennt.

Diese Zeile bewirkt einen Rücksprung zu Zeile 40, der Zeile hinter GOSUB.

Nicht die RETURN-Zeile vergessen, sonst gibt es Fehler!



In BASIC wird das Wort GOSUB verwendet, wenn der Computer ein Unterprogramm abarbeiten soll. Am Ende des Unterprogramms steht das Wort RETURN. Im Anschluß an GOSUB sollte die Nummer der ersten Zeile des Unterprogramms stehen. (RETURN braucht keine Zeilennummer.) Der Computer springt automatisch zum Hauptprogramm zurück und bearbeitet die Stelle, die an der Reihe gewesen wäre, bevor er das Hauptprogramm verlassen hat. Ein Unterprogramm kann an jeder Stelle des Programms aufgerufen werden, sooft es gewünscht wird.

GOSUB-Programme

Ein Unterprogramm ist dann nützlich, wenn eine bestimmte Aufgabe an verschiedenen Stellen des Programms öfter wiederholt werden soll. Hier sind noch weitere Programme mit Unterprogrammen.

Zahlen-Programm

```
50 INPUT A
60 INPUT B
70 GOSUB 250
80 PRINT "A GETEILT DURCH B=";A/B
90 GOTO 50

250 REM UNTERPROGRAMM STOP
260 IF A=0 AND B=0 THEN STOP
270 RETURN
```

Dieses Unterprogramm ermöglicht es, aus dem Programm herauszugehen. Wenn der Divisionsvorgang gestoppt werden soll, gibt man 0 in den Zeilen 50 und 60 ein. Das Wort STOP braucht in diesem Programm nicht vor dem Unterprogramm zu stehen, da Zeile 90 für einen Rücksprung sorgt.

Umwandlungs-Programm

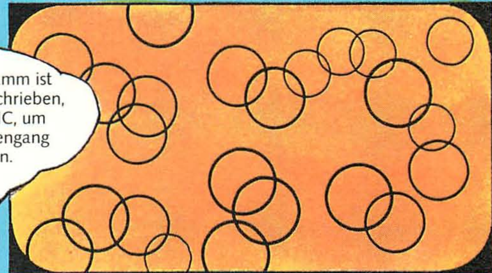
```
100 INPUT "ENTFERNUNG";KM
110 INPUT "ZEIT";H
120 GOSUB 200
130 PRINT "DURSCHNITTSGESCHWINDIGKEIT"
140 PRINT KM/H;" KM/H UND";M/H;" M/H"
150 STOP
200 REM        UMWANDLUNG KM IN MEILEN
210 LET M=KM/1.609
220 RETURN
```

Dieses Unterprogramm verwandelt Kilometer in Meilen. Man kann ein Unterprogramm in vielen verschiedenen Programmen verwenden, muß allerdings darauf achten, daß immer die gleichen Variablen-Namen benutzt werden.

Kreis-Programm

```
1 MITTELPUNKT = X,Y
2 RADIUS = R
3 FARBE = X
4 GOSUB 10
5 GOTO 1
10 REM        KREIS ZEICHNEN
11 ZEICHNE KREIS MIT MITTEL-
   PUNKT X,Y;RADIUS R; FARBE X
12 RETURN
```

Dieses Programm ist in Deutsch geschrieben, nicht in BASIC, um den Gedankengang zu zeigen.



Unterprogramme sind für Grafik-Programme nützlich, wenn Diagramme gezeichnet werden sollen mit Zahlen, die im Hauptprogramm errechnet worden sind. Mit diesem Programm könnte man viele verschiedene Kreise zeichnen, wenn man in den Zeilen 1 bis 5 entsprechende Informationen eingibt.

Quiz-Programm

```
5 LET C=0
10 PRINT "WANN WURDE ERFUNDEN?"
20 READ C$,F
30 PRINT C$
40 INPUT A
50 LET C=C+1
60 IF C=3 THEN STOP
70 GOSUB 100
80 GOTO 10

100 REM ANTWORTEN
110 IF ABS(A-F)<10 THEN PRINT "STIMMT"
120 IF ABS(A-F)>10 THEN PRINT "FALSCH"
130 PRINT "NÄCHSTER VERSUCH"
140 RETURN

200 DATA TELEFON,1876,DRUCKERPRESSE,1450,FAHRRAD,1791
```

In Zeile 20 sucht der Computer die Datenzeile und speichert das erste Wort in C$ und die erste Zahl in F.

Ausdruck des Wortes aus C$.

Der Zähler C stoppt das Programm, wenn es dreimal gelaufen ist, weil sich nur drei Datenwörter in C$ und F befinden.

Dies ist das Unterprogramm.

Jedesmal, wenn das Programm wiederholt wird, werden die Wörter und Zahlen in C$ und F durch das nächste Datenpaar ersetzt.

Dieses Programm verwendet ein Unterprogramm zur Überprüfung der Antworten auf die Fragen. Die richtigen Antworten sind in F gespeichert, und die Antworten der befragten Person gehen nach A. In den Zeilen 100 und 110 des Unterprogramms vergleicht der Computer A mit F. Das Wort ABS bedeutet „absoluter Wert"; es bewirkt, daß der Computer die Differenz zwischen den Zahlen A und F feststellt. (Dabei werden Minuszahlen nicht beachtet.) Ist die Differenz kleiner als 10, gibt der Computer OK aus. Ist die Differenz größer als 10, gibt er NEIN aus.

Spielereien mit Wörtern

Die meisten Computer können die Wörter, die in Variablen gespeichert sind, überprüfen und verschiedene Dinge damit anfangen. So kann der Variablen-Inhalt daraufhin überprüft werden, ob ein bestimmtes Wort oder ein bestimmter Buchstabe enthalten ist. Dies ist dann nützlich, wenn jemand, der das Programm anwendet, Worteingaben überprüfen will. Computer können auch Buchstaben oder Wörter in anderer Reihenfolge zusammensetzen und sie an Buchstaben in anderen Variablen anhängen. Hier wird gezeigt, wie man dies alles in BASIC machen kann.

1

```
10 A$="ICH BIN DUMM"
20 B$="NUR NARREN DENKEN"
30 C$=B$+" "+A$
40 PRINT C$
RUN
NUR NARREN DENKEN ICH BIN DUMM
```

Bei den meisten Mikrocomputern kann man den Begriff LET weglassen – allerdings nicht beim ZX81.

2

```
PRINT LEFT$(B$,3)
NUR
PRINT LEFT$(B$,3)+" "+A$
NUR ICH BIN DUMM
```

Man kann den Inhalt zweier Variablen zusammensetzen wie im Beispiel oben. Der Zwischenraum in Anführungszeichen bewirkt, daß zwischen den Wörtern ein Zwischenraum erscheint.

Man kann auch Teile von Variablen zusammenfügen wie im Beispiel oben. LEFT$(B$,3) bedeutet: Nimm die ersten drei Buchstaben von links aus B$.

3

```
PRINT RIGHT$(A$,4)
DUMM
```

RIGHT$ mit dem Namen der Zeichenkette und der Anzahl der gewünschten Buchstaben bedeutet in diesem Fall: Nimm die ersten 4 Buchstaben von rechts.

4

```
PRINT MID$(B$,5,4)
NARR
```

Dies veranlaßt den Computer, mittlere Buchstaben zu nehmen. Die erste Zahl berechnet den ersten Buchstaben, die zweite Zahl sagt, wie viele Buchstaben gewünscht werden.

Beim Zählen der Buchstaben müssen auch Zwischenräume und Satzzeichen mitgezählt werden.

5

```
10 K$="BIMM BAMM"
20 PRINT LEN(K$)
RUN
9
```

Um festzustellen, wie lang eine Zeichenkette ist – Anzahl der Buchstaben, Zwischenräume und Symbole –, verwendet man LEN (Kurzform von engl. length = Länge).

WENN A$ = "COMPUTERBUCH", was ist LEFT$(A$,8)? RIGHT$(A$,10)? MID$(A$,5,3)?

Anmerkung für Benutzer von Sinclair-Computern

```
PRINT A$(9 TO 12)
DUMM
PRINT B$(5 TO 8)
NARR
```

Das bedeutet: Nimm die Buchstaben 9 bis 12.

Der Sinclair-Computer kennt die Anweisungen LEFT$, RIGHT$ und MID$ nicht, aber mit der Methode oben kann man den Computer beliebige Buchstaben wählen lassen.

Programm "Geheimcode"

Mit diesem Programm kann man Wörter verschlüsseln. Ähnliche, allerdings viel kompliziertere Programme benutzen auch Geheimdienste, um Codes zu schaffen oder zu knacken.
Am leichtesten versteht man dieses Programm, wenn man eine geheime Nachricht auf ein Blatt Papier schreibt, dann die Programmzeilen durcharbeitet und das tut, was der Computer mit der Nachricht macht; das sollte man aufschreiben.

```
5 LET C$=""
7 LET D$=""
10 PRINT "NACHRICHT EINGEBEN"
20 INPUT M$
30 PRINT "GEHEIMZAHL ZWISCHEN
   2 UND";LEN(M$)-1
40 INPUT N
50 LET A$=RIGHT$(M$,N)
60 LET B$=LEFT$(M$,LEN(M$)-N)
70 LET M$=A$+B$
80 FOR I=1 TO LEN(M$) STEP 2
90 LET C$=C$+MID$(M$,I,1)
100 NEXT I
110 FOR J=2 TO LEN(M$) STEP 2
120 LET D$=D$+MID$(M$,J,1)
130 NEXT J
140 LET M$=C$+D$
150 PRINT "GEHEIMNACHRICHT"
160 PRINT M$
170 END
```

Leere String-Variablen werden bereitgestellt.

Dies bedeutet: Länge der Nachricht minus 1.

N (die geheime Zahl) Buchstaben vom rechten Teil von M$.

Die Länge von M$ minus N Buchstaben vom linken Teil von M$, d. h. der Rest der Buchstaben.

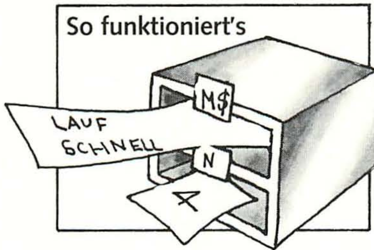
Ersetzt die Buchstaben in M$ durch A$+B$.

Von 1 bis zur Buchstabenanzahl der Nachricht in Zweierschritten, d. h. 1, 3, 5 usw. Bei jeder Wiederholung der I-Schleife wird in Zeile 90 ein Buchstabe aus Position I von M$ nach C$ gebracht.

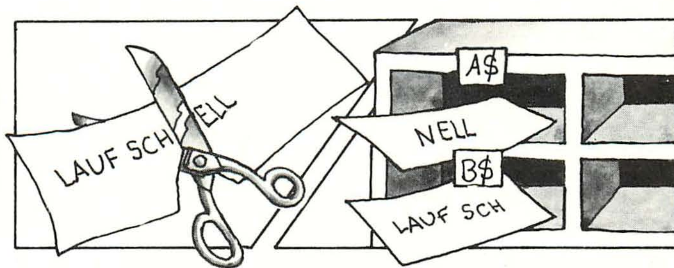
Von 2 bis zur Buchstabenanzahl der Nachricht in Zweier-Schritten, d. h. 2, 4, 6 usw. Funktioniert wie die I-Schleife.

Ersetzt nochmals die Buchstaben in M$.

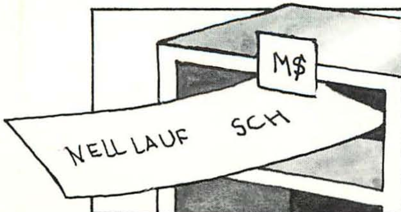
So funktioniert's



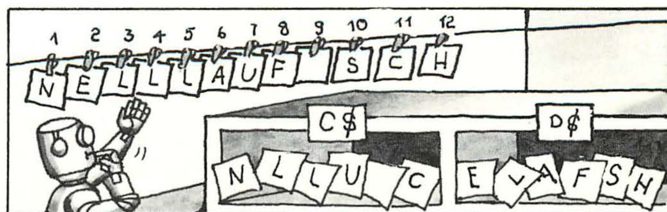
Angenommen, die Nachricht heißt „Lauf schnell", und die Geheimzahl ist 4. Diese Angaben werden in M$ und N gespeichert.



In den Zeilen 50 und 60 benutzt der Computer die Geheimzahl zur Aufteilung der Nachricht. In Zeile 50 werden die vier rechten Buchstaben der Nachricht genommen und in A$ gespeichert. In Zeile 60 wird der Rest der Nachricht in B$ gespeichert.



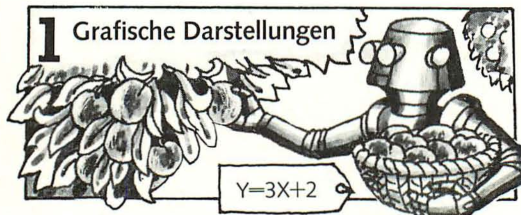
In Zeile 70 werden A$ und B$ zusammengesetzt. Die Buchstaben vom Ende der Nachricht werden nach vorn gesetzt.



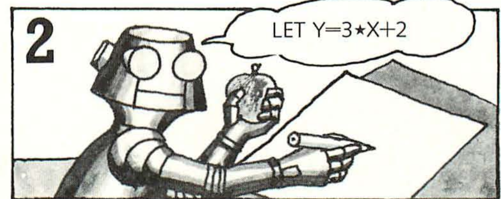
Jedesmal, wenn die I-Schleife durchlaufen wird, wird ein Buchstabe in C$ gespeichert, der sich an ungerader Stelle befindet (also N, L, L usw.). Jedesmal, wenn die J-Schleife durchlaufen wird, wird ein Buchstabe in D$ gespeichert, der sich an gerader Stelle befindet (also E, L, A usw.). Dann werden C$ und D$ zur verschlüsselten Nachricht zusammengesetzt.

Graphen und Symbole

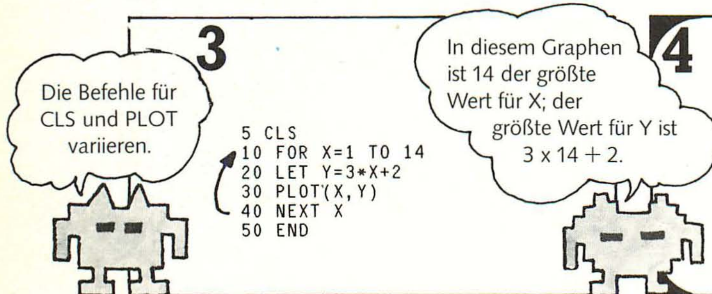
Man kann einen Computer so programmieren, daß er Informationen auf verschiedene Weise ausgibt, z. B. als Wörter, Zahlen, Bilder oder Grafiken. Durch die Verwendung von Grafiken, Bildern und Symbolen kann man schwierige Informationen leichter verständlich machen.



Man stelle sich einen Pfirsichbaum vor, dessen Fruchternte entsprechend seinem Alter von Jahr zu Jahr wächst. Das kann man mit einer Gleichung ausdrücken, z. B. Y = 3X + 2 (Y ist die Ernte, X ist das Alter). Dies kann man sich schwer vorstellen, die grafische Darstellung (ein Graph) ist daher hilfreich.



Mit einem Computer ist es sehr einfach, das Wachsen der Ernte Y in bezug auf das Alter X grafisch darzustellen. Um den Graphen zu zeichnen, muß man den Wert für Y für jeden Wert von X finden. Dies liefert auf einfache Weise die Anweisung LET Y = 3*X+2.





So sieht das Programm für diesen Graphen aus: Die Schleife setzt X auf alle Werte zwischen 1 und 4. Bei jedem Durchgang durch die Schleife verwendet die Zeile 20 den gerade gültigen Wert für X, um Y zu berechnen. Zeile 30 zeichnet X und Y auf den Bildschirm. Bei Programmen für das Zeichnen von Graphen muß man sicherstellen, daß die maximalen Werte für X und Y auf dem Bildschirm liegen, andernfalls gibt es einen Fehler.

Computer und Mathematik

Bei Rechnungen, die wie 3 x X + 2 aus verschiedenen Teilen bestehen, führt der Computer immer zuerst die Multiplikationen und Divisionen aus, bevor er addiert oder subtrahiert. Der Computer würde also bei den folgenden Aufgaben jeweils übereinstimmende Ergebnisse ausgeben.

```
PRINT 4*6+8          PRINT 8+4*6
32                   32
```

Soll der Computer die Aufgabe in anderer Reihenfolge ausrechnen, dann benutzt man Klammern, z. B. so:

```
PRINT (8+4)*6
72
```

Jetzt addiert der Computer 8 und 4 und multipliziert das Ergebnis mit 6.

Programmierübung

```
MAN DENKE SICH EINE ZAHL,
VERDOPPELE SIE, ADDIERE 4,
DIVIDIERE DURCH 2, ADDIERE 7,
MULTIPLIZIERE MIT 8, SUBTRAHIERE 12,
DIVIDIERE DURCH 4 UND SUBTRAHIERE 11.
WIE HEISST DAS ERGEBNIS?
DIE GEDACHTE ZAHL IST ...
```

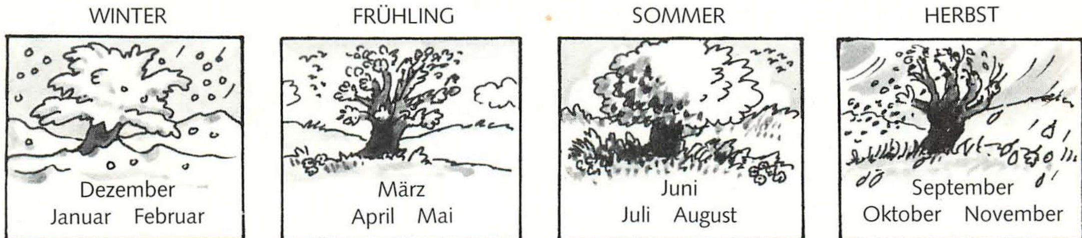
Schreiben Sie ein Programm für diesen bekannten Zahlentrick. (Man bestimmt die gedachte Zahl, indem man vom Rechenergebnis 4 abzieht und die dann erhaltene Zahl durch 2 teilt.)

Geburtstags-Programm

Dieses Programm veranschaulicht eine andere Art der Darstellung von Informationen auf dem Bildschirm. Es benutzt Symbole, um die Anzahl von Menschen, die in den verschiedenen Jahreszeiten geboren sind, zu vergleichen. Ein solches Programm kann z. B. das Auftreten einer bestimmten Vogelart in verschiedenen Jahreszeiten vergleichen oder die Anzahl der gewonnenen Spiele verschiedener Fußballmannschaften. Bei einem Programm dieser Länge empfiehlt es sich, zunächst einen Programmplan anzufertigen.

Programmplan

Ziel: Vergleich der Anzahlen von Menschen, die im Winter, Frühling, Sommer und Herbst geboren wurden.



1. Eingabe der Daten (d. h. der Jahreszeiten, in denen die Menschen geboren wurden).
2. Speicherung der Daten im Computer.
3. Ausgabe der Daten auf dem Bildschirm.

Programm-Beispiel

```
RUN
IM WINTER   : *******
IM FRÜHLING: ***
IM SOMMER   : *****
IM HERBST   : *****
```

Das Programm

```
5 LET A=0
6 LET B=0
7 LET C=0
8 LET D=0
10 FOR I=1 TO 20
20 PRINT "PERSON ";I;" GEBOREN IM"
30 PRINT "WINTER, FRÜHLING, SOMMER ODER HERBST?"
40 PRINT "DRUCKE W, F, S ODER H"
50 INPUT B$
60 IF B$="W" THEN LET A=A+1
70 IF B$="F" THEN LET B=B+1
80 IF B$="S" THEN LET C=C+1
90 IF B$="H" THEN LET D=D+1
100 NEXT I
110 PRINT "IM WINTER   : ";
115 LET N=A
120 GOSUB 200
130 PRINT "IM FRÜHLING: ";
135 LET N=B
140 GOSUB 200
150 PRINT "IM SOMMER   : ";
155 LET N=C
160 GOSUB 200
170 PRINT "IM HERBST   : ";
175 LET N=D
180 GOSUB 200
190 STOP
200 REM UNTERPROGRAMM STERNE
210 IF N=0 THEN GOTO 250
220 FOR I=1 TO N
230 PRINT "*";
240 NEXT I
250 PRINT
260 RETURN
```

Zeilen 5–8: Leere Variablen für die Gesamtzahl für jede Jahreszeit.

Zeile 10: Schleife zur Befragung jeder einzelnen Person.

Die Zeilen 60 bis 90 prüfen die Antwort in B$ und addieren 1 zur Variablen der betreffenden Jahreszeit.

Zeile 100: Schickt den Computer zurück zur weiteren Fragestellung.

Zeile 120: Das Unterprogramm bewirkt, daß der Computer so viele Sterne ausdruckt, wie die Anzahl in jeder Variablen ist.

Zeilen 135–155: Durch die Speicherung der jeweiligen Gesamtzahl in N kann dasselbe Unterprogramm für jede Jahreszeit verwendet werden.

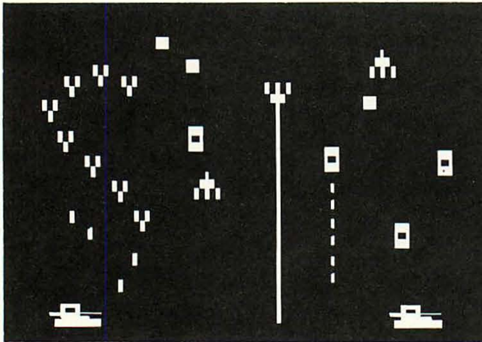
Zeile 170: Bewirkt das Ausdrucken der Sterne in derselben Zeile.

Zeile 210 ist für den Fall da, daß niemand in einer bestimmten Jahreszeit geboren wurde.

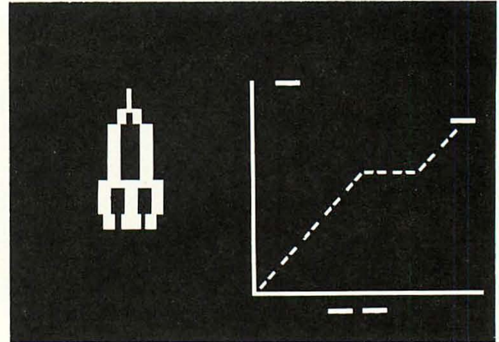
Zeile 220: Im Hauptprogramm wird N gleich der Gesamtzahl für A, B, C oder D gesetzt. Die Schleife bewirkt, daß der Computer die Zeile 230 „N"-mal ausführt.

Bewegte Bilder

Auf diesen beiden Seiten wird gezeigt, wie man PLOT und UNPLOT anwenden kann, um auf dem Bildschirm bewegte Bilder zu erzeugen. Bewegte Bilder nennt man *Animationsgrafik.* Sie sind nützlich für Spiele oder zur Illustration von Programmen, die z. B. das Schwerkraftprinzip oder Flugbahnen erläutern.



Bei Video- und Computerspielen wird die Grafik von einem kleinen Computer gesteuert. Der Computer ist so programmiert, daß er nur diese Spiele beherrscht. Die Programme sind nicht in BASIC geschrieben, sondern im Code des Computers.



Ein Mikrocomputer, der in BASIC „für den Hausgebrauch" programmiert ist, kann nur einfachere und langsamere Bilder erzeugen. Er kann die Befehle an den Bildschirm nicht rasch genug bewältigen, um wirklich schnell bewegte Grafik zu zeichnen.

1 Zeichen-/Lösch-Programm

```
10 LET X=1
20 LET Y=1
30 PLOT(X,Y)
40 UNPLOT(X,Y)
50 LET X=X+1
60 LET Y+Y+1
70 GOTO 30
```

Manche Computer brauchen eine Programmzeile für den Grafik-Modus.

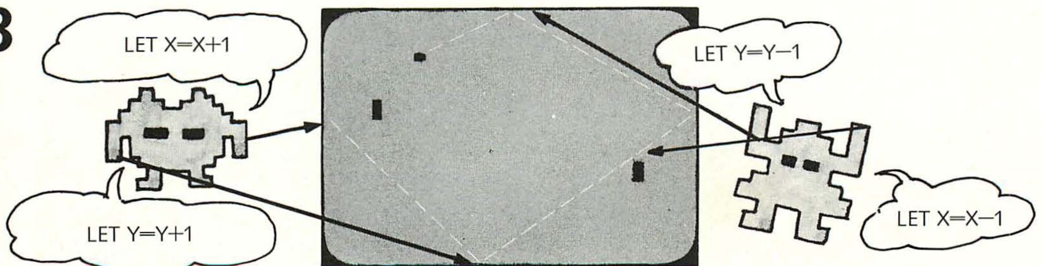


Dieses kurze Programm läßt einen Lichtpunkt über den Bildschirm wandern. Denken Sie daran, daß die Befehle für PLOT und UNPLOT bei den verschiedenen Computerarten unterschiedlich sind.



Sobald der Punkt eine Ecke des Bildschirms erreicht, stoppt das Programm mit einer Fehlermeldung, da sich die Werte von X und Y außerhalb der Bildschirmfläche des Computers befinden.

3

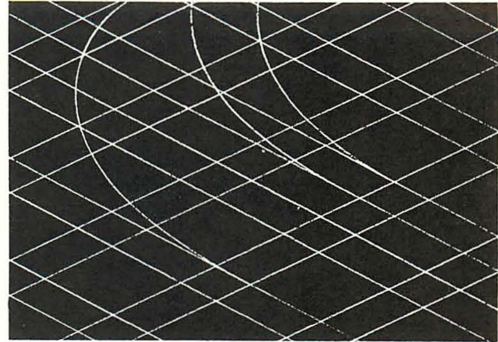


Video-Feldspiele verwenden Programme wie das oben gezeigte, um den Ball auf dem Bildschirm zu bewegen. Es gibt einfache Programmregeln, die den Ball in Bewegung halten, sobald er den Bildschirmrand erreicht hat.

Berührt der Ball den oberen Bildrand, dann wird der zu Y zu addierende Betrag statt dessen abgezogen. Analog wird bei Berührung des rechten Bildrandes der Betrag von X abgezogen.

Programm für Linienmuster

Dieses Programm zeichnet eine Linie über den Bildschirm. Wenn die Linie einen Rand erreicht, wird sie in einer anderen Richtung weitergeführt. UNPLOT wird nicht verwendet, daher hinterlassen die Linien ein Muster auf dem Bildschirm. Das Bild rechts zeigt das Ergebnis nach dem Programmstart. Zeile 100 des Programms bewirkt die Zeichnung von 10 000 Bildpunkten. Man kann diese Anzahl verändern, um den Vorgang zu verkürzen; man kann auch BREAK benutzen, um das Programm abzubrechen, wenn ein gewünschtes Muster erreicht ist.



```
10 REM FALLS NOTWENDIG HIER ANWEISUNG FÜR GRAFIK-MODUS
20 PRINT "ZAHL DER BILDPUNKTE BREITE?";
30 INPUT B
40 PRINT "ZAHL DER BILDPUNKTE HÖHE?";
50 INPUT H
55 CLS
```

Die Zeilen 20 bis 50 fragen nach Höhe und Breite des Bildschirms. Das Semikolon stellt die Antwort in dieselbe Zeile wie die Frage.

```
60 LET X=B/2
70 LET Y=H/2
```

Dies läßt X und Y im Bildschirm-Mittelpunkt beginnen.

```
80 LET S=1
90 LET T=1
```

S und T sind die Größen, die zu X und Y addiert werden, damit die Linie sich bewegt.

```
100 FOR I=1 TO 10000
```

Die Schleife von Zeile 100 bis 190 wird 10 000mal wiederholt. Jedesmal werden X und Y um einen winzigen Betrag verändert.

```
110 LET S=S+(INT(RND(1)*10+1)-5)/50
120 LET X=X+S
130 LET Y=Y+T
```

Dies bewirkt, daß bei jedem Schleifendurchlauf ein Betrag zu X addiert wird, der um einen kleinen Betrag schwankt.

```
140 IF X<5 THEN LET S=-S
150 IF X>B-5 THEN LET S=-S
160 IF Y<5 THEN LET T=-T
170 IF Y>H-5 THEN LET T=-T
```

Diese Zeilen sind der Randtest. S und T ändern die Richtung, sobald X und Y sich bis auf 5 Bildpunkte einem Rand nähern.

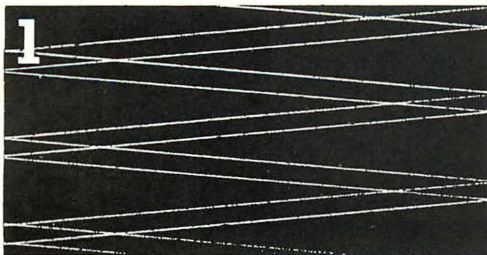
```
180 GOSUB 300
190 NEXT I
200 STOP
300 REM PLOT LINE
```

Hier wird der Computer zum Unterprogramm des Linienzeichnens geschickt.

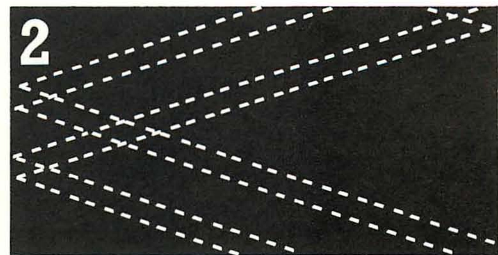
```
310 PLOT(X,Y)
320 RETURN
```

Zeichnet den Bildpunkt mit dem gerade gültigen Wert für X und Y.

Experimente



Zeile 110 addiert zu X jedesmal eine winzige Zufallszahl. Daher „wackelt" die Linie über den Bildschirm. Wenn Sie einen Computer haben, versuchen Sie, diese Linie zu löschen. Die Linien auf dem Bildschirm sollten parallel laufen.



Versuchen Sie, die Zahlen in Zeile 80 und 90 zu verändern, z. B. 5 oder 10 (oder noch größer bei einem Computer mit hochauflösender Grafik). Dadurch entstehen unterbrochene Linien.

Verse aus dem Computer

Auf den folgenden Seiten wird ein Programm vorgestellt, mit dem man eine Reihe von *„Gedichten"* erzeugen kann. Eine Fassung dieses Programms ist im *„Ersten Buch der Elektronik"* (Otto Maier Verlag, 1983) erschienen. Dort wird der Bau eines „Papier-Computers" gezeigt. Das gleiche Programm wird hier in einer BASIC-Version vorgestellt.

Programm für den Papier-Computer

6. DREHE DEN KREISEL, UM N ZU FINDEN.

Papier-Computer

Zahlenkreisel

```
1   A=0 und B=0
2   Erhöhe A um 1
3   Wenn A=6 gehe zu Zeile 10
4   Schreibe Datenzeile A
5   Erhöhe B um 1
6   Drehe Kreisel und ermittle N
7   Schreibe Datenwörter aus Zeile B,
    Spalte N
8   Wenn B=3 oder 5, gehe zu Zeile 5
9   Gehe zu Zeile 2
10  Stop
```

Datenzeilen

```
ES WAR EINMAL EIN MANN AUS
DER
SEIN
DOCH DANN IN DER NACHT
ER FAND NIEMALS HERAUS
```

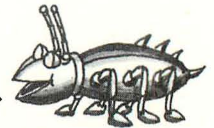
Datenwörter

1	Groß Rist	Bad Nist	Neu Wist	Klein Quist
2	versteckte	bemalte	zeigte	bedeckte
3	Bein	Hemd	Haus	Schaf
4	in Batist	als Tourist	Das war Mist	und tanzt Twist
5	flog es weg	wurd' es blau	platzte es	fiel es um
6	daß es kracht	mit Bedacht	und es lacht	grad' um acht
7	wo es ist	was es frißt	was es ist	was ihr wißt

```
Es war einmal ein Mann aus Neu Wist
der versteckte sein Schaf als Tourist
doch dann in der Nacht
wurd' es blau und es lacht
er fand niemals heraus
```

Dies ist das Programm für den Papier-Computer. Es sieht so aus, als sei es in BASIC geschrieben, aber es läuft nicht auf einem echten Computer. Wörter und Sätze sind auf Papierstückchen geschrieben; das Programm weist an, welche davon auszuwählen sind. Der Zahlenkreisel liefert Zufallszahlen zwischen 1 und 4.

1 Übertragung des Programms in BASIC



So läuft es noch nicht auf dem Computer.

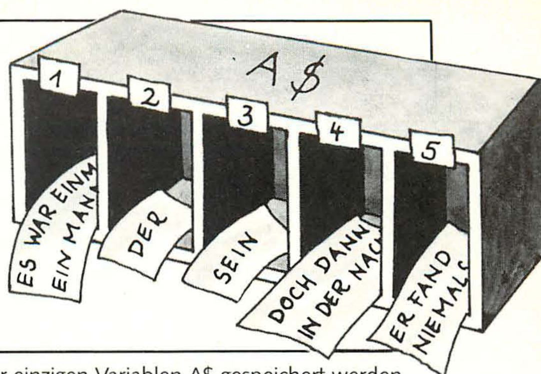
```
10 LET A=0
20 LET B=0
30 LET A=A+1
40 IF A=6 THEN STOP
50 Schreibe Datenzeile A
60 LET B=B+1
70 LET N=INT(RND(1)*4+1)
80 Schreibe Datenwörter aus Zeile B,
   Spalte N
90 IF B=3 THEN GOTO 60
100 IF B=5 THEN GOTO 60
110 GOTO 30
120 END
```

- Zeilen 10 und 20: Diese Zeilen setzen die Variablen gleich Null.
- Zeilen 30 und 40: Die Zeilen 30 und 40 zählen die vom Computer ausgewählten Datenzeilen.
- Zeilen 50 und 80: Die Zeilen 50 und 80 sind noch nicht in BASIC.
- Zeile 60: Zeile 60 zählt die Datenwörter.
- Zeile 70: Erzeugt eine Zufallszahl zwischen 1 und 4.
- Zeilen 90 und 100: Die Zeilen 90 und 100 setzen zurück, um eine andere Datenzeile auszuwählen.

Der größte Teil des Programms läßt sich leicht in BASIC übertragen; schwieriger ist das mit den Zeilen 50 und 80. Der Computer muß einen Weg finden, um aus den Datenzeilen die entsprechenden Wörter für jede Zeile des „Gedichts" auszuwählen.

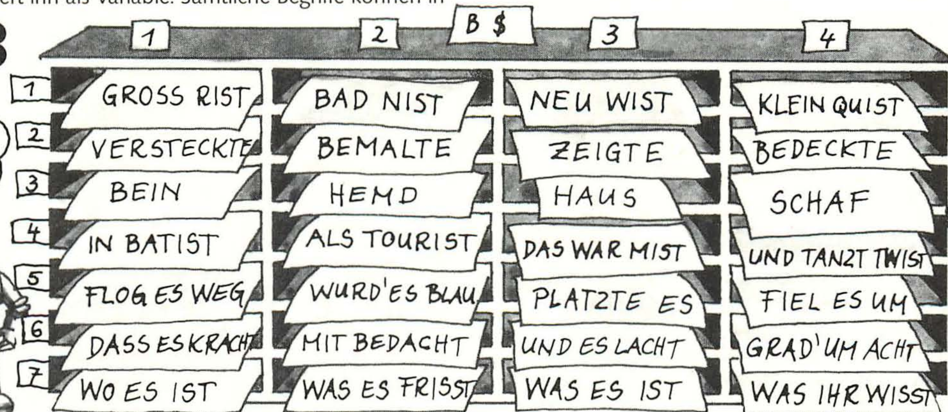
2 Der Computer wird gefüttert

```
50 READ A$
.
.
180 DATA ES WAR EINMAL EIN MANN AUS,
    DER,SEIN
190 DATA DOCH DANN IN DER NACHT,
    ER FAND NIEMALS HERAUS
```



Um den Computer mit den Datenzeilen und Datenwörtern zu versorgen, wird die Anweisung READ ... DATA verwendet. Jedesmal wenn der Computer die READ-Anweisung ausführt, nimmt er einen neuen Begriff aus der Datenzeile und speichert ihn als Variable. Sämtliche Begriffe können in einer einzigen Variablen A$ gespeichert werden. Eine Variable, die mehr als einen Begriff enthält, bezeichnet man als *Feld.* Jeder Begriff wird durch eine Nummer identifiziert. So ergibt z. B. READ A$(3) den Begriff „sein"*.

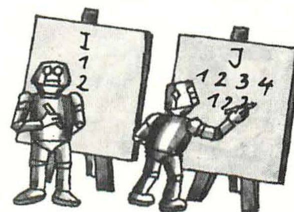
3

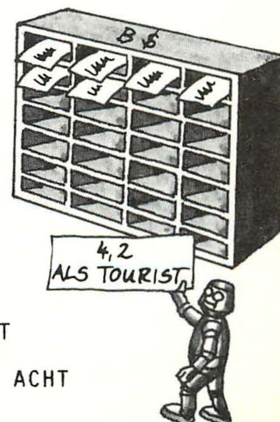


Eine Variable kann mehrere Datenreihen umfassen. Alle Daten können in einer Variablen abgelegt werden, wie das Bild zeigt. Man spricht dann von einem *zweidimensionalen Feld.* Jeder Begriff wird durch die Reihennummer und die Spaltennummer identifiziert. So ergibt READ B$(4, 2) „als Tourist" und READ B$(6, 3) „und es lacht". Zahlen können als Variablen ebenfalls in Feldern abgelegt werden.

4 Daten in Variablen umwandeln

```
10 FOR I=1 TO 7 }—— I ist die Reihennummer.
20 FOR J=1 TO 4 }—— J ist die Spaltennummer.
30 READ B$(I,J)
40 NEXT J
50 NEXT I
60 DATA GROSS RIST,BAD NIST,NEU WIST,KLEIN QUIST
70 DATA VERSTECKTE, BEMALTE,ZEIGTE,BEDECKTE
80 DATA BEIN,HEMD,HAUS,SCHAF
90 DATA IN BATIST,ALS TOURIST,DAS WAR MIST,UND TANZT TWIST
100 DATA FLOG ES WEG,WURD' ES BLAU, PLATZTE ES,FIEL ES UM
110 DATA DASS ES KRACHT,MIT BEDACHT,UND ES LACHT,GRAD' UM ACHT
120 DATA WO ES IST,WAS ES FRISST,WAS ES IST,WAS IHR WISST
```





Damit die Daten eingelesen werden können, müssen die Zahlen in Klammern nach dem READ-Befehl geändert werden. Dies kann mittels einer geschachtelten Schleife geschehen. B$ verwendet im Beispiel oben eine I-Schleife für die Reihennummer und eine J-Schleife für die Spaltennummer. Jedesmal wenn die I-Schleife ausgeführt wird, wird die J-Schleife viermal wiederholt – einmal für jede Spalte einer Reihe.

* Sinclair-Computer behandeln Variablen auf eine andere Art, daher läuft dieses Programm nicht auf einem Sinclair. Mehr darüber auf der nächsten Seite.

5 Speicherplatz für Variablen

```
5 DIM K$(5)
10 FOR I=1 TO 5
20 READ K$(I)
30 NEXT I
40 STOP

60 DATA HUND,KATZE,FLOH,WANZE
```

Dimensionierung der Variablen: 5 Begriffe in einer Reihe.

Mit dieser Anweisung werden die Daten in K$ gespeichert, sooft die Schleife wiederholt wird.



Am Anfang des Programms muß man dem Computer sagen, wieviel Speicherraum er für die Variablen zu reservieren hat. Der DIM-Befehl, gefolgt vom Namen der Variablen und der Zahl der Elemente, z. B. DIM K$(5), *dimensioniert* den Speicherplatz.

Für ein zweidimensionales Feld werden die Anzahl der Reihen und der Spalten angegeben, z. B. DIM C$(5, 3). Die Zahlen müssen der Anzahl der Elemente entsprechen, sonst wird ein Fehler gemeldet.

6 Daten ausdrucken

```
200 LET A=0
210 LET B=0
220 LET A=A+1
230 IF A=6 THEN STOP
240 PRINT A$(A)
250 LET B=B+1
260 LET N=INT(RND(1)*4+1)
270 PRINT B$(B,N)
280 IF B=3 THEN GOTO 250
290 IF B=5 THEN GOTO 250
300 GOTO 220
310 END
```

A zählt, wie oft dieser Teil des Programms wiederholt wird.

B zählt die Reihen mit den Wörtern und stellt sicher, daß immer die entsprechende Reihe verwendet wird.

Die Zeilen 280 und 290 bewirken, daß Wörter aus einer anderen Datenreihe angezeigt werden, bevor die nächste Datenzeile ausgedruckt wird.

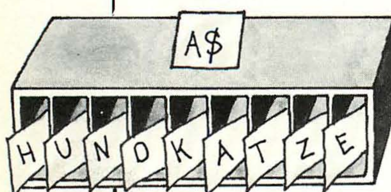
Diese Zeile veranlaßt den Computer zurückzugehen, um die nächste Datenzeile anzuzeigen.

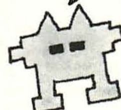
Der Computer benötigt diese Anweisungen, um die Datenzeilen und Datenwörter in der richtigen Reihenfolge anzuzeigen. Dieser Programmteil wird fünfmal wiederholt. Dabei werden jedesmal die Daten der Zeile A und einige Wörter der Reihe B angezeigt. Durch die Zufallszahl N wird festgelegt, welche Wörter ausgewählt werden.

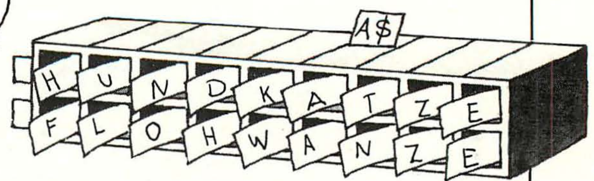
Sinclair-Computer und Variablen

Das Programm läuft in dieser Form nicht auf Sinclair-Computern, da dort Zeichenketten auf eine andere Art verarbeitet werden.

Bei zweidimensionalen Feldern müssen sowohl die Nummer der Reihe als auch die Nummern der Buchstaben angegeben werden; z. B. A$(2, 5 TO 9) entspricht „Wanze".







Damit ein Sinclair-Computer weiß, welche Datenelemente er aus dem Datenfeld auswählen soll, muß man ihm die Nummern des ersten und letzten Buchstabens des gewünschten Begriffs angeben. Mit dieser Methode werden bei Sinclair-Computern auch die Anweisungen LEFT$, RIGHT$ u. a. durchgeführt (siehe Seite 78).

Am Anfang des Programms wird angegeben, wieviel Reihen das Feld hat und wieviel Buchstaben in jeder Reihe sind. DIM A$(2, 9) bedeutet: 2 Reihen mit jeweils 9 Buchstaben. Alle Reihen des Feldes müssen die gleiche Anzahl Buchstaben haben.

Das vollständige Programm

Jetzt können alle Teile des Programms zusammengefaßt und eingetippt werden. Im ersten Abschnitt des Programms (Zeilen 10 bis 190) werden die Daten eingelesen, im zweiten Abschnitt (Zeilen 200 bis 310) wird das Gedicht angezeigt. Jedesmal wenn das Programm gestartet wird, ergibt sich wegen der Zufallszahl N ein neues Gedicht, da der Computer sich andere Wörter aussucht.

```
10 DIM A$(5)
20 DIM B$(7,4)
30 FOR I=1 TO 7
40 FOR J=1 TO 4
50 READ B$(I,J)
60 NEXT J
70 NEXT I
80 DATA GROSS RIST,BAD NIST,NEU WIST,KLEIN QUIST
90 DATA VERSTECKTE, BEMALTE,ZEIGTE,BEDECKTE
100 DATA BEIN,HEMD,HAUS,SCHAF
110 DATA IN BATIST,ALS TOURIST,DAS WAR MIST,UND TANZT TWIST
120 DATA FLOG ES WEG,WURD' ES BLAU, PLATZTE ES,FIEL ES UM
130 DATA DASS ES KRACHT,MIT BEDACHT,UND ES LACHT,GRAD' UM ACHT
140 DATA WO ES IST,WAS ES FRISST,WAS ES IST,WAS IHR WISST
150 FOR I=1 TO 5
160 READ A$(I)
170 NEXT I
180 DATA ES WAR EINMAL EIN MANN AUS,DER,SEIN
190 DATA DOCH DANN IN DER NACHT,ER FAND NIEMALS HERAUS

200 LET A=0
210 LET B=0
220 LET A=A+1
230 IF A=6 THEN STOP
240 PRINT A$(A)
250 LET B=B+1
260 LET N=INT(RND(1)*4+1)
270 PRINT B$(B,N)
280 IF B=3 THEN GOTO 250
290 IF B=5 THEN GOTO 250
300 GOTO 220
310 END
```

Die Zeilen 10 und 20 weisen den Speicherplatz für die Variablen an: 5 Buchstaben für A$ und 7 Reihen zu 4 Buchstaben für B$.

Das sind die verschachtelten Schleifen zum Einlesen der Daten in B$.

In den Zeilen 80 bis 140 stehen alle Wörter, die im Feld B$ zu speichern sind.

Schleife zum Einlesen der Daten in A$.

In den Zeilen 180 und 190 stehen alle Wörter, die im Feld A$ zu speichern sind.

Zeigt die Datenzeile an, die in A$ unter A gespeichert ist.

Zeigt die Wörter an, die in B$ Reihe B, Spalte N gespeichert sind.

Das Programm endet mit Zeile 230, wenn A=6; es kommt also nie zu Zeile 310. Manche Computer benötigen aber als Abschluß den Befehl END.

Beispiele

```
ES WAR EINMAL EIN MANN AUS
KLEIN QUIST
DER
BEDECKTE
SEIN
HEMD
DAS WAR MIST
DOCH DANN IN DER NACHT
FLOG ES WEG
MIT BEDACHT
ER FAND NIEMALS HERAUS
WAS IHR WISST
```

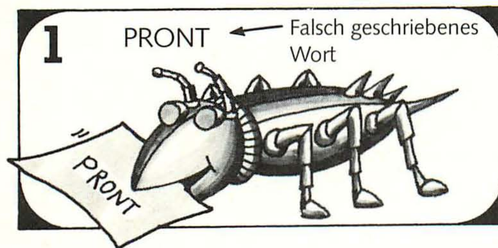
```
ES WAR EINMAL EIN MANN AUS
BAD NIST
DER
BEMALTE
SEIN
HAUS
IN BATIST
DOCH DANN IN DER NACHT
PLATZTE ES
DASS ES KRACHT
ER FAND NIEMALS HERAUS
WO ES IST
```

Hier sind zwei Versionen der 16 384 möglichen Gedichte. Wenn immer wieder die gleichen Gedichte angezeigt werden, sollte man im Handbuch nachlesen, wie der Computer unterschiedliche Zufallszahlen erzeugt. Manche Computer erzeugen nämlich nach dem Einschalten jedesmal die gleiche Folge von Zufallszahlen.

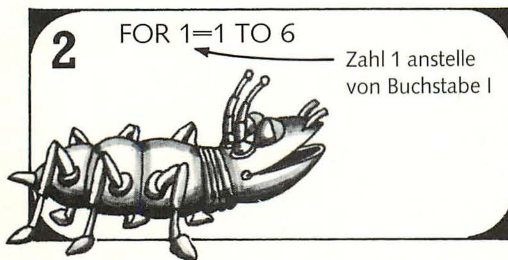
Hinweise zum Programmieren

Auf diesen beiden Seiten sind einige Hinweise zusammengestellt, die das eigene Programmieren erleichtern. Außerdem ist eine Liste der häufigsten Fehler und ihrer Ursachen aufgeführt. Die wahrscheinlichsten Fehler kommen zuerst. Sollte ein Programm also nicht laufen, dann empfiehlt es sich, das Programm auf mögliche Fehler zu überprüfen.

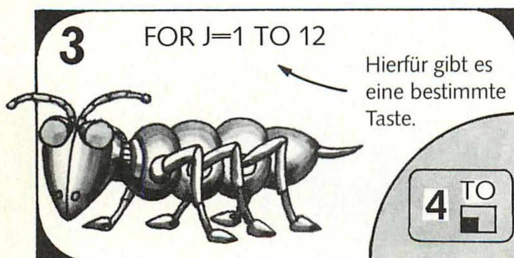
Fehlersuche



Zunächst sollten BASIC-Ausdrücke auf Tippfehler untersucht werden. BASIC-Ausdrücke mit Tippfehlern kann der Computer nicht erkennen.



Überprüfen Sie alle O (Buchstabe) und 0 (Ziffer Null) sowie I (Buchstabe) und 1 (Ziffer), um sicherzustellen, daß die richtigen Zeichen am richtigen Platz stehen.



Beim Sinclair-Computer müssen die vorhandenen Tasten für die BASIC-Wörter benutzt werden; die Wörter werden also nicht Buchstabe für Buchstabe eingetippt.

Programme schreiben

Beim Schreiben von Programmen sollte man immer an die drei Haupttätigkeiten eines Computers denken: Ein Computer kann einfache Anweisungen ausführen, Vorgänge wiederholen und Entscheidungen treffen. Diese sind die Bausteine aller Programme.




```
LET A=3
LET N=N+1
PRINT A/T
PLOT(X,Y)
```




```
FOR J=1 TO 6

20 LET A=1
30 IF A<10 THEN
   GOTO 100
```



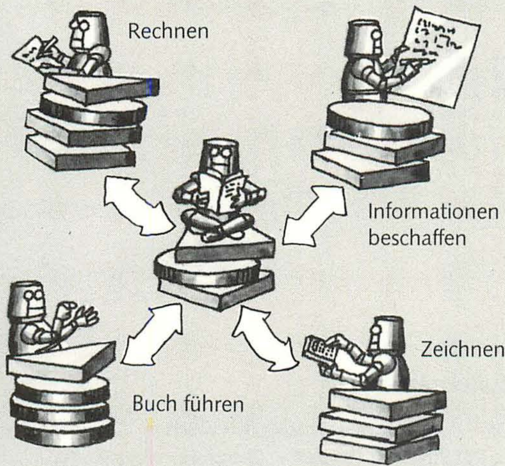

```
IF X=Y THEN STOP
IF K$="HALLO"
  THEN PRINT A
```

In diesem Buch sind die wesentlichsten BASIC-Befehle behandelt, die genügen, um den Computer diese Tätigkeiten ausführen zu lassen. Beim Schreiben von Programmen sollte man überlegen, was der Computer an einem bestimmten Punkt tun muß; danach richtet sich dann die Entscheidung über die richtigen Befehle.

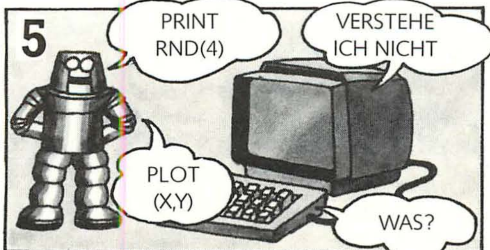


Achten Sie auch immer auf fehlende Anführungszeichen oder fehlende Kommas zwischen Datenangaben. Schwierige Zeilen, die viele Symbole und Zeichen enthalten, sollten Sie besonders sorgfältig prüfen.

Normalerweise gibt es mehrere Möglichkeiten, ein Programm zu schreiben; manche sind vielleicht kürzer und „eleganter" als andere. Schreibt man ein langes Programm, empfiehlt es sich, das Programm zu unterteilen und Unterprogramme für jede Aufgabe einzubauen. Das Kernprogramm kann aus einfachen Anweisungen, Entscheidungen und Wiederholungen bestehen und bestimmen, wann und wie oft der Computer die Unterprogramme ausführen soll.



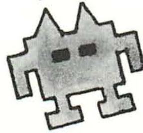
Eine solche Aufteilung von Programmen in einzelne Bereiche erleichtert die Fehlersuche. Dabei kann jeder Bereich getrennt überprüft werden, ohne daß man das ganze Programm ablaufen lassen muß. Man sollte auch daran denken, jeden Bereich mit einer REM-Zeile zu kennzeichnen, die seinen Zweck angibt.



Vergewissern Sie sich, daß die RND-, PLOT- und CLS-Befehle richtig eingegeben werden. Überprüfen Sie ferner, ob Sie dem Computer eine allgemeine Grafik-Zeile eingegeben haben, falls er eine braucht.

Fehlermeldungen

Alle Computer geben bei fehlerhaften Programmen Fehlermeldungen. Diese Meldungen sind im Computer-Handbuch erklärt. Hier sind einige der häufigsten Fehlermeldungen zusammengestellt.



◀ Dies bedeutet, daß in der DATA-Zeile nicht ausreichend viele Daten für den Computer zu lesen sind. Die Ursache dafür kann z. B. ein fehlendes Komma zwischen zwei Daten sein, so daß der Computer diese zwei Daten als ein Datum auffaßt.

▶ Eine Zeile mit dieser in einer GOTO- oder GOSUB-Anweisung angegebenen Nummer existiert nicht. Die Ursache dafür kann ein Tippfehler sein, oder Sie haben vielleicht die Zeile gelöscht, indem Sie eine neue Zeile mit der gleichen Nummer eingegeben haben.





◀ Diese Anzeige kann z. B. bei einem Sinclair-Computer erscheinen. Sie bedeutet, daß eine bestimmte Variable nicht in einer Zeile wie in LET C=0 oder LET C="" definiert wurde.

▶ Dies bedeutet, daß in einer Schleife die NEXT-Zeile fehlt. Ursache kann ein falsch eingegebener Variablen-Name sein, den der Computer nicht erkennt, oder ein Tippfehler, z. B. I statt 1.



Verflixt noch mal

Manche Fehler sind sehr schwer zu finden. Wenn das Programm jedoch nicht läuft, muß irgendwo ein Fehler stecken. Ist der Fehler unauffindbar, empfehlen wir, schwierige Zeilen nochmals einzugeben. Beim zweiten Mal werden sie dann vielleicht doch richtig, ohne daß man merkt, wo der Fehler nun eigentlich gesteckt hat.

Lösungen

Seite 61

Programm "Name und Nachricht"

```
10 PRINT "WIE HEISSEN SIE?"
20 INPUT N$
30 PRINT "HALLO"
40 PRINT N$
50 PRINT "WIE GEHT ES IHNEN?"
```

Seite 63

1. Rechenprogramm

```
10 LET A=9
20 LET B=7
30 PRINT A*B
40 PRINT A/B
50 LET A=A+1
60 LET B=B+3
70 PRINT A*B,A/B
80 END
```

Komma für Zwischenraum

2. Einmaleins-Programm

Leerstellen

```
30 PRINT A;" MAL ";B;" IST ";A*B
40 PRINT A;" GETEILT DURCH ";B;"IST ";A/B
```

3. Programm "Name und Nachricht"

(Abwandlung)

```
10 PRINT "WIE HEISSEN SIE?"
20 INPUT N$
30 PRINT " HALLO ";N$;" WIE GEHT ES IHNEN?"
```

Seite 64

Rechenprogramm

```
10 PRINT "WIEVIEL IST 7 MAL 7?"
20 INPUT A
30 IF A=49 THEN PRINT "STIMMT!"
40 IF A<>49 THEN PRINT "NEIN!";7*7
```

Das Semikolon an dieser Stelle ist wichtig!

Seite 65

Alter-Rate-Programm

Man ersetze Zeile 30 und füge eine neue Zeile 35 hinzu:

```
30 IF G<14 THEN PRINT "ÄLTER ALS"
35 IF G>14 THEN PRINT "JÜNGER ALS"
```

Seite 69

Grafik-Programm

```
5 LET C=0
45 LET C=C+1
50 IF C<6 THEN GOTO 10
```

Gezeichnete Initialen

Beispielprogramm für das Zeichnen des Buchstabens L.

```
10 LET X=15
20 LET Y=30
30 PLOT (X,Y)
40 LET Y=Y-1
50 IF Y>5 THEN GOTO 30
60 LET X=X+1
70 PLOT (X,Y)
80 IF X<45 THEN GOTO 60
90 END
```

Seite 70

Zufallszahlen

Die Formel für eine Zufallszahl zwischen 10 und 20 ist INT(RND(1)*11+9). Bei Computern, bei denen eine Zahl in Klammern hinter RND genügt, wäre es RND(11)+9. Da es elf Zahlen zwischen 10 und 20 gibt, muß man Zufallszahlen von 1 bis 11 wählen und 9 hinzuzählen.

Seite 71

Angriff im Weltraum

Hier sind die Zeilen, die man hinzufügen muß, um die Trefferzahl zu zählen.

```
15 LET S=0
75 IF X=A*B THEN LET S=S+1
95 PRINT "GETROFFEN ";S;" VON 6 GEGNERN"
```

Seite 73

1. Achterreihen-Programm

```
10 PRINT "ACHTERREIHE"
20 FOR J=1 TO 12
30 PRINT J;"* 8 =";J*8
40 NEXT J
```

Seite 73

2. 1malN-Reihe

```
10 INPUT "ZAHL EINGEBEN: ";N
20 PRINT "ES FOLGT DIE ";N;"-REIHE"
30 FOR I=1 TO 12
40 PRINT I;" MAL ";N;" IST ";I*N
50 NEXT I
60 INPUT "NEUE ZAHL EINGEBEN (J/N)";M$
70 IF M$="J" THEN GOTO 10
```

Beim ZX81 benötigt man getrennte PRINT- und INPUT-Zeilen.

Seite 78

Computerbuch-Worträtsel

```
LEFT$(A$,8) ist "COMPUTER"
RIGHT$(A$,10) ist "PUTER BUCH"
MID$(A$,5,8) ist "UTER BUC"
```

Seite 80

Zahlentrick-Programm

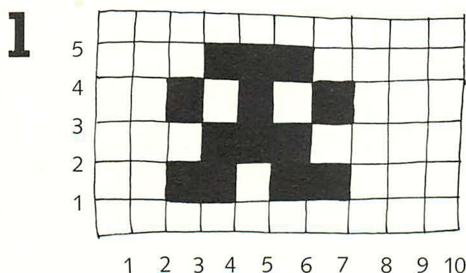
```
10 PRINT "MAN DENKE SICH EINE ZAHL,"
20 PRINT "VERDOPPELE SIE, ADDIERE 4,"
30 PRINT "DIVIDIERE DURCH 2, ADDIERE 7,"
40 PRINT "MULTIPLIZIERE MIT 8, SUBTRAHIERE 12,"
50 PRINT "DIVIDIERE DURCH 4 UND SUBTRAHIERE 11."
60 PRINT "WIE HEISST DAS ERGEBNIS?"
70 INPUT N
80 PRINT "DIE GEDACHTE ZAHL IST";(N-4)/2
```

Die Klammern sind für die richtige Rechenreihenfolge wichtig.

Seite 75

Raumfahrer-Programm

1



Hier ist ein kleiner „Raumfahrer" auf kariertes Papier gezeichnet.

2

4,5; 5,5; 6,5

3,4; 5,4; 7,4

4,3; 5,3; 6,3

3,2; 4,2; 6,2; 7,2

Dies sind die Koordinaten derjenigen Quadrate, aus denen die Figur besteht.

3

```
5 CLS
50 INPUT "WIEVIELE PUNKTE IN DER BREITE";B
60 INPUT "WIEVIELE PUNKTE IN DER HÖHE";H
65 CLS
70 FOR I=0 TO H STEP H/6
80 FOR I=0 TO B STEP B/6



130 NEXT J
140 NEXT I
150 END
```

Man kann die 6 auch in eine größere Zahl abändern, dann erscheinen entsprechend mehr Raumfahrerfiguren auf dem Bildschirm. (Ist die Zahl jedoch zu groß, gibt es einen Fehler.)

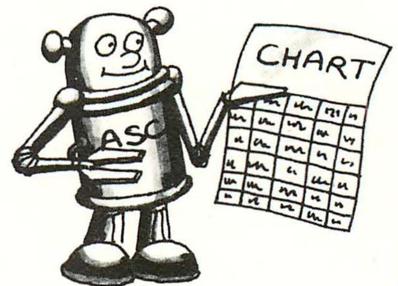
Die Zeichen-Zeilen gehören hierhin, z. B.
90 PLOT (J+3, I+2)
92 PLOT (J+4, I+2),
für den Fall der beiden Quadrate unten links in der Figur oben. Für jedes Quadrat ist eine Programmzeile erforderlich.

Schreiben Sie das Muster-Wiederholungsprogramm ab, und zwar ohne die Zeilen 10 bis 40 und 90 bis 140, wie oben gezeigt. (Diese Zeilen stellen das Zufallsmuster des Programms her, werden also nicht benötigt.)
Fügen Sie nun Ihre eigenen Zeichen-Zeilen in das Programm ein, und zwar zwischen die Zeilen 80 und 140. (Sie können die Programmzeilen auch neu numerieren.)
Um den „Raumfahrer" zu wiederholen, muß man bei jedem Koordinaten-Paar zur ersten Zahl J und zur zweiten Zahl I hinzuzählen.

Kleines BASIC-ABC

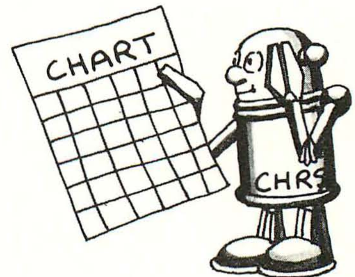
Auf den folgenden Seiten sind die wichtigsten Ausdrücke zusammengestellt und erklärt, die Sie zum Programmieren in BASIC brauchen. Die meisten kommen auch in Programmen vor, die Sie in diesem Buch finden, so daß Sie an der entsprechenden Stelle nachschauen können, was sie in einem Spiel bewirken. Allerdings kann man nicht alle Befehle auf allen Heimcomputern benutzen. Die Vergleichstabelle auf Seite 96 zeigt Ihnen, was Sie auf Ihrem Gerät verwenden können.

ABS (Abkürzung für absolut): Läßt den Computer einen Zahlenwert absolut nehmen, ohne Rücksicht darauf, welches Vorzeichen die Zahl hat; z. B. ist ABS (–10) ebenso 10 wie ABS (+10).

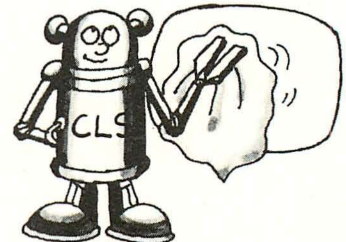


ASC: Setzt ein Zeichen in die entsprechende ASCII-Codezahl um; ASC("3") ergibt z. B. 51. Der in Klammern stehende Ausdruck muß aus einer Zeichenkette bestehen, z. B. ASC(A$) oder ASC("20").
ZX81 (Timex 1000) und Sinclair Spectrum (Timex 2000) arbeiten nicht mit ASC, obwohl der Spectrum den ASCII-Code benutzt; für diese Typen heißt der entsprechende Befehl CODE, gefolgt von einer Zeichenkette.

**CHR$**: Wandelt Zahlen in Buchstaben um. Mit Ausnahme des ZX81 (Timex 1000) benutzen dazu alle Computer, die in diesem Buch genannt sind, den ASCII-Code, in dem jedes Zeichen einer bestimmten Zahl entspricht. So entspricht z. B. der Buchstabe A der Codezahl 65, und der Befehl PRINT CHR$(65) läßt ein A auf dem Bildschirm erscheinen.
Man kann CHR$ in Verbindung mit INT und RND benutzen, um den Computer Zufallsbuchstaben wählen zu lassen, z. B. so: CHR$(INT(RND*26+65)). Dieser Befehl erzeugt Zufallsbuchstaben auf einem Sinclair Spectrum. Den entsprechenden Befehl für andere Geräte finden Sie in der Vergleichstabelle auf Seite 96.



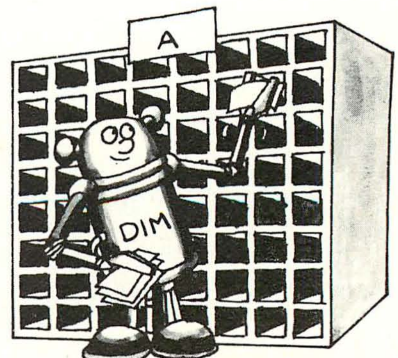
CLS (engl. **cl**ear **s**creen): Läßt alles verschwinden, was sich auf dem Bildschirm befindet, ohne daß der Speicherinhalt gelöscht oder etwas daran geändert wird. Dieser Befehl kann z. B. dazu dienen, ein Programmlisting vom Bildschirm zu löschen, bevor man das Programm mit RUN ablaufen läßt, oder um in einem Computerspiel dem Mitspieler ein Bild nur für eine begrenzte Zeit zu zeigen.
Apple und Commodore benutzen HOME bzw. PRINT CHR$(147) anstelle von CLS.



CODE: siehe **ASC**.

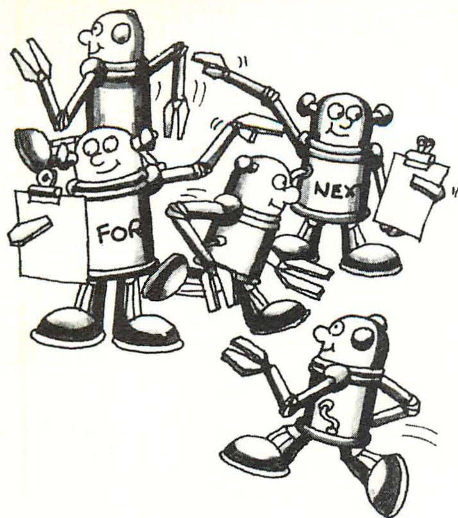
DATA (Daten): Liste von Dingen, wie Wörter oder Zahlen, die vom Computer in Variablen gespeichert werden. Siehe dazu auch **READ**.

DIM (Dimension): Sagt dem Computer, wieviel Speicherraum er für eine indizierte Variable zur Verfügung stellen soll. DIM X(6) bedeutet z. B., daß eine Reihe für sechs Elemente zur Verfügung gestellt und diese Reihe mit X benannt werden soll. DIM A(5,4) bedeutet, daß für die Variable A fünf Reihen zu je vier Spalten benötigt werden.



EDIT: Ermöglicht die Abänderung einer Programmzeile, ohne daß die gesamte Zeile nochmals eingegeben werden muß.

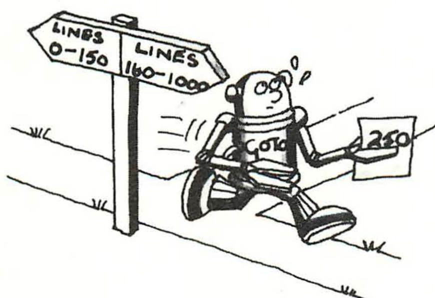
END (Ende): Teilt dem Computer das Programmende mit. Bei manchen Computern ist der END-Befehl stets erforderlich, andere Computer, wie z. B. Sinclair, benötigen diesen Befehl nicht.



FOR...NEXT: Mit FOR beginnt eine Schleife, die den Computer einen Teil des Programms eine bestimmte Anzahl von Malen wiederholen läßt. Nach FOR muß eine Variable folgen (z. B. V für die Anzahl der Versuche in einem Spiel), und diese Variable muß einen Anfangs- und einen Endwert angeben (z. B. 1 TO 10).
Das Ende der Schleife wird bestimmt durch NEXT (im Beispiel oben NEXT V). Dadurch wird der Wert der Variablen jedesmal um 1 erhöht, und der Computer springt wieder in die FOR-Zeile zurück. Ist der Endwert der Variablen erreicht, übergeht der Computer die NEXT-Zeile und fährt mit der darauf folgenden Programmzeile fort. Auf ein FOR muß also immer ein NEXT folgen.

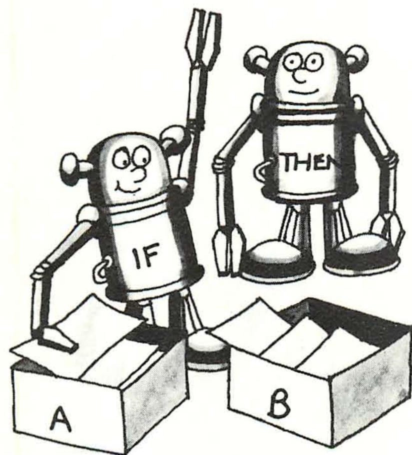
GET: siehe **INKEY$**.

GOSUB (engl. Abkürzung für **go** to **sub**routine = gehe zum Unterprogramm): Veranlaßt den Computer, das Hauptprogramm zu verlassen und ein Unterprogramm zur Bewältigung einer speziellen Aufgabe abzuarbeiten. Auf GOSUB muß die Nummer der ersten Zeile des Unterprogramms folgen. Am Ende des Unterprogramms muß der Computer ins Hauptprogramm zurückgeschickt werden, und zwar durch den Befehl RETURN mit der Nummer der gewünschten Programmzeile. Auf ein GOSUB muß also immer ein RETURN folgen.



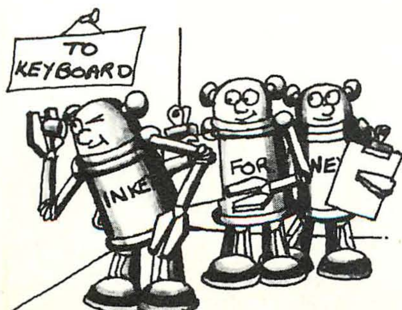
GOTO (gehe zu...): Läßt den Computer in eine andere Programmzeile springen, ohne daß die Zeilen, die auf dem Weg dorthin liegen, abgearbeitet werden. Auf GOTO muß natürlich die Nummer der Programmzeile folgen, in die der Computer springen soll.

HOME: siehe **CLS**.

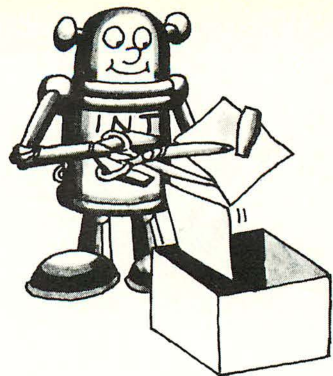


IF...THEN (wenn... dann): Veranlaßt den Computer zu prüfen, ob eine Entscheidung richtig oder falsch ist, oder einzelne Daten (z. B. Zahlen, Wörter oder Variablen-Inhalte) miteinander zu vergleichen und bestimmte Dinge zu veranlassen, die vom Ergebnis der Entscheidung bzw. des Vergleichs abhängen. IF... THEN wird mit den folgenden Zeichen sowie mit AND oder OR benutzt:
= gleich / genausogroß / viel wie
< weniger / kleiner als
> mehr / größer als
<= weniger / kleiner als oder gleich / genausogroß / viel wie
>= mehr / größer als oder gleich / genausogroß / viel wie
<> anders als / nicht identisch mit
Wenn der Computer zu dem Ergebnis kommt, daß ein Ausdruck richtig ist bzw. der Vergleich positiv ist, folgt er der Anweisung, die mit THEN gegeben ist. Andernfalls befolgt er diese Anweisung nicht und macht mit der nächsten Zeile weiter.

INKEY$: Prüft die Tastatur, um festzustellen, ob eine Taste gedrückt wurde und gegebenenfalls welche. Hier kommt es nicht darauf an, daß man eine Taste drücken soll wie bei INPUT.
INKEY$ wird üblicherweise in einer Schleife gebraucht, die den Computer immer wieder veranlaßt, die Tastatur zu prüfen. Computer arbeiten nämlich so schnell, daß man in der Zeit, die der Computer braucht, um einen Prüfimpuls auszuschicken, noch nicht einmal die Taste drücken könnte.
Apple und VC-20 arbeiten nicht mit INKEY$; der entsprechende Befehl für den VC-20 lautet GET X$, für den Apple siehe die Vergleichstabelle auf Seite 96.



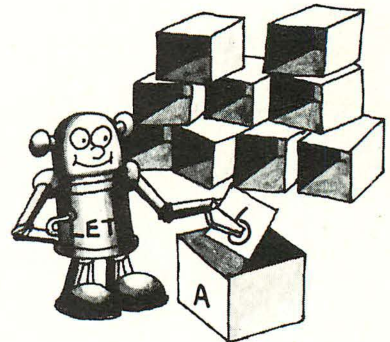
INPUT (Eingabe): Bezeichnet einen Speicherplatz, gibt ein Fragezeichen aus und wartet darauf, daß Sie etwas eingeben, was in den Speicher gebracht werden soll. Das Programm läuft nicht weiter, bevor Sie nicht RETURN, ENTER oder NEWLINE drücken.



INT (engl. Abkürzung für **int**eger = ganze Zahl): Verwandelt eine positive Dezimalzahl (Kommazahl) in eine ganze Zahl, indem alle Ziffern rechts vom Komma (Dezimalpunkt) weggelassen werden, z. B. INT(20.999) = 20. Bei negativen Dezimalzahlen läßt INT alle Ziffern rechts vom Komma verschwinden und erhöht die erste Ziffer links vom Komma um 1, z. B. INT(–3.6) = 4.
INT wird häufig in Verbindung mit RND gebraucht, z. B. so: INT(RND*20+1) sagt dem Computer, daß er eine beliebige ganze Zahl zwischen 1 und 20 auswählen soll.

**LEFT$** (engl. left = links): Bewirkt, daß der Computer sich mit dem linken Teil eines Wortes befaßt. LEFT$(A$,4) bedeutet z. B.: Nimm von links die ersten 4 Buchstaben von A$.

LEN (engl. Abkürzung für length = Länge): Gibt die Länge einer Zeichenkette an, d. h. die Anzahl der Buchstaben in einer Variablen.



LET (es sei): Kennzeichnet einen Speicherplatz und belegt ihn mit einem bestimmten Wert. LET A=6 bedeutet z. B., daß ein Speicherplatz mit "A" bezeichnet und mit dem Wert 6 belegt wird. Man nennt das „einer Variablen (hier A) einen Wert (hier 6) zuweisen".

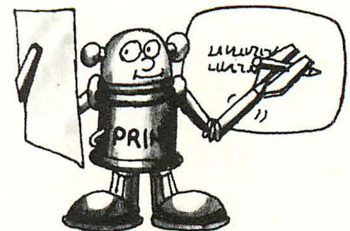
LIST (auflisten): Läßt das Programm auf dem Bildschirm erscheinen.

MID$ (engl. Abkürzung für **mid**dle = Mitte): Bewirkt, daß der Computer sich mit dem mittleren Teil eines Wortes befaßt.
MID$(A$,4,3) bedeutet z. B.: Nimm 3 Buchstaben von A$, beginnend mit dem vierten Buchstaben.

NEW (neu): Löscht das Programm aus dem Speicher des Computers, um Platz zu machen für ein neues.

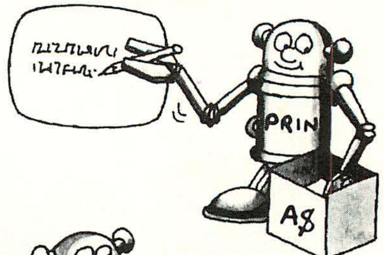
NEWLINE (Zeilenwechsel): Sagt dem Computer, daß eine Programmzeile oder eine Eingabe vollständig eingegeben wurde. Bei manchen Computern heißt diese Taste RETURN oder ENTER.

PEEK (schau nach): Damit kann man feststellen, welchen Wert eine bestimmte Speicherzelle enthält. Man benutzt PEEK mit einer Nummer, die die Adresse der betreffenden Speicherzelle darstellt. Der Acorn/BBC arbeitet nicht mit PEEK.

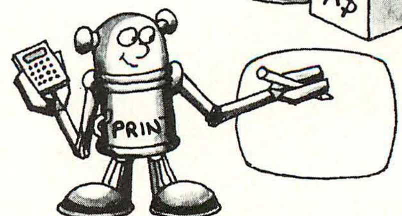


PLOT (zeichne): Veranlaßt den Computer zum Zeichnen eines Bildpunktes; z. B. bedeutet PLOT (X,Y): Zeichne den Bildpunkt mit den Koordinaten X (nach rechts) und Y (nach oben).

POKE (schiebe hinein): Damit kann man in eine Speicherzelle einen bestimmten Wert hineinschreiben. Wie bei PEEK geschieht das unter Angabe der Speicheradresse.
Der Acorn/BBC arbeitet nicht mit POKE.



PRINT (drucke): Läßt den Computer etwas auf den Bildschirm bringen. Dieser Befehl läßt sich auf verschiedene Arten nutzen:
Mit PRINT kann man eine beliebige Nachricht auf dem Bildschirm erscheinen lassen; sie braucht nicht in BASIC formuliert zu sein, sie muß lediglich in Anführungszeichen stehen.
PRINT gefolgt von einem Variablennamen, z. B. PRINT A oder PRINT A$, läßt den Computer den Inhalt dieser Variablen auf den Bildschirm bringen.
Mit PRINT kann man auch Rechnungen durchführen und das Ergebnis auf dem Bildschirm erscheinen lassen. Gibt man z. B. PRINT 6*4 ein, liest man auf dem Bildschirm 24.
PRINT ohne jeden Zusatz bewirkt eine Leerzeile.

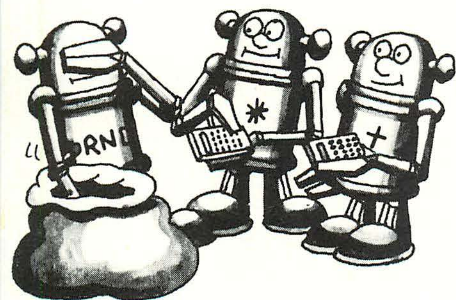


READ (lies): Veranlaßt den Computer, die Information in einer DATA-Zeile zu lesen und in einer Variablen zu speichern. Vergleiche **DATA**.

READY (fertig, bereit): Manche Computer zeigen dadurch an, daß sie bereit sind, weitere Anweisungen auszuführen.

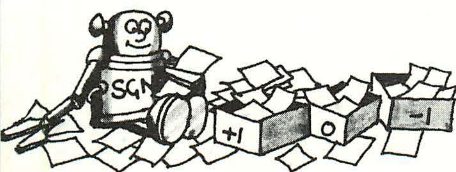
REM (engl. Abkürzung für **rem**ark = Bemerkung): REM-Zeilen dienen dazu, Programmteile mit einer entsprechenden Bemerkung inhaltlich zu kennzeichnen. Der Computer ignoriert Zeilen, die mit REM beginnen, er gibt sie jedoch im Programmlisting aus.

RETURN (kehre zurück): Bewirkt am Ende eines Unterprogramms, daß der Computer zum Hauptprogramm zurückkehrt und die Stelle bearbeitet, die an der Reihe gewesen wäre, bevor er das Hauptprogramm verlassen hat. Vergleiche **GOSUB**.



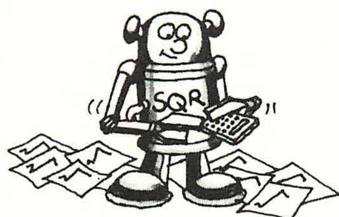
**RIGHT$** (engl. right = rechts): Bewirkt, daß der Computer sich mit dem rechten Teil eines Wortes befaßt. RIGHT$(A$,4) bedeutet z. B.: Nimm von rechts die ersten 4 Buchstaben von A$.

RND (engl. Abkürzung für **rand**om = zufällig): Läßt den Computer eine Zufallszahl auswählen. Die verschiedenen Computertypen benutzen RND unterschiedlich, wie aus der Vergleichstabelle auf Seite 96 zu ersehen ist. Sinclair-Computer z. B. erzeugen mit RND ohne jeden Zusatz eine Zahl zwischen 0 und 0,99999999. Diese Begrenzung läßt sich überschreiten, wenn man RND mit einer Zahl multipliziert und eine Zahl hinzuzählt. So erzeugt RND*20 eine Zahl zwischen 0 und 19,99999999 und RND*20+1 eine Zahl zwischen 1 und 20,99999999.
Zur Erzeugung ganzer Zahlen siehe **INT**.
Zur Erzeugung von Zufallszahlen und -symbolen siehe **CHR$**.



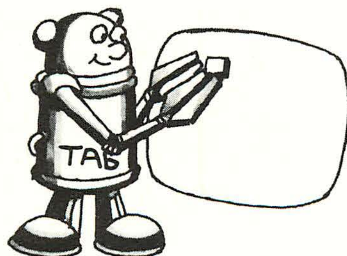
RUN (lauf): Bewirkt, daß der Computer das Programm ausführt.

SGN (engl. Abkürzung für **sign** = Vorzeichen): Veranlaßt den Computer, das Vorzeichen einer Zahl anzugeben. Für eine negative Zahl zeigt er –1 an, für null 0 und für eine positive Zahl +1; z. B. führt SGN(–30) zur Anzeige –1, SGN(7) zu +1 und SGN(0) zu 0.



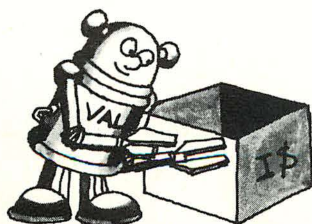
SQR (engl. Abkürzung für **sq**uare **r**oot = Quadratwurzel):
Bewirkt, daß der Computer die Quadratwurzel aus einer Zahl zieht.

STEP (Schritt): Wird bei FOR...NEXT-Schleifen verwendet und gibt an, mit welchem neuen Wert der Zahlen-Variablen wiederholt werden soll.



STOP (halt): Läßt den Computer im Programm anhalten. (Bei den meisten Computern kann man statt dessen auch END verwenden, mit Ausnahme des Sinclair ZX81.)

TAB (Liste): Läßt den Cursor auf dem Bildschirm zu einer bestimmten Spalte wandern. TAB wird in der Regel zusammen mit PRINT benutzt, um etwas in der Mitte des Bildschirms erscheinen zu lassen. Die Anzahl der Stellen, die der Cursor weitergehen soll, wird hinter TAB in Klammern aufgeführt. Die größtmögliche Zahl hängt von der Bildschirmbreite Ihres Geräts ab.



UNPLOT (Gegenteil von PLOT): Bewirkt das Löschen eines Bildpunktes.

VAL (engl. Abkürzung für **val**ue = Wert): Läßt den Computer den numerischen Wert von Zahlen nehmen, die als Zeichenkette geschrieben sind. Das bedeutet praktisch, daß der Computer die Zeichenkette als Zahlen-Variable behandelt; wenn z. B. I$ = "60", dann ergibt VAL(I$) die Zahl 60.

Vergleichstabelle

Diese Tabelle zeigt einige Unterschiede im BASIC, wie sie bei bestimmten Computertypen vorkommen. Sie enthält allerdings keine Anweisungen für Grafik, Ton oder Farbe, da diese sich von Gerät zu Gerät stark unterscheiden. Beachten Sie auch, daß die meisten Heimcomputer (mit Ausnahme des Acorn/BBC) zwar mit PEEK und POKE arbeiten, aber die Speicheradressen in unterschiedlicher Weise bezeichnen, so daß die Zahlen, die auf PEEK oder POKE folgen, für jeden Computertyp entsprechend geändert werden müssen.

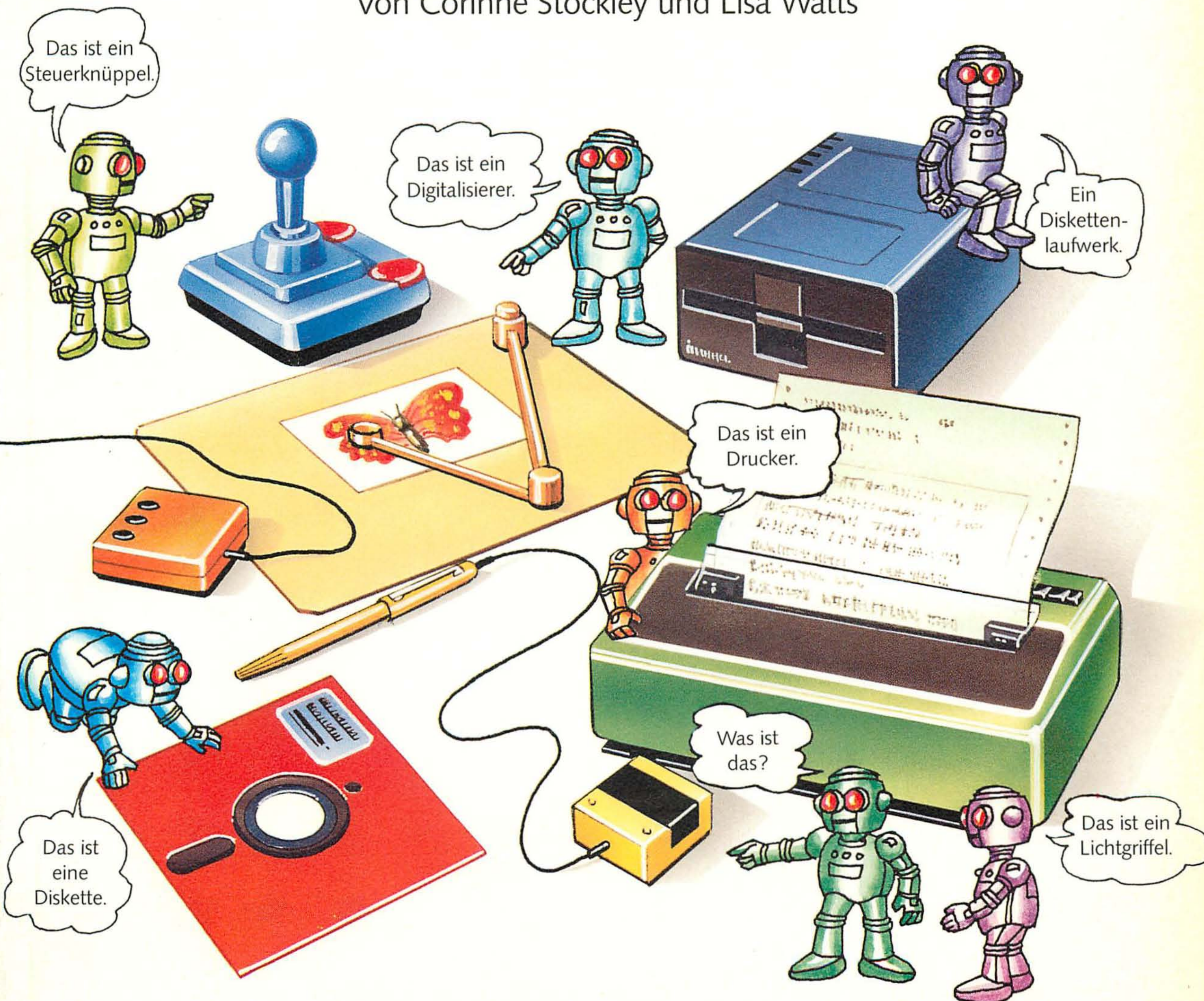
	Acorn/BBC	**Commodore Pet/VC-20**	**Apple**	**TRS-80**	**Sinclair Spectrum**	**Sinclair ZX81**
Wähle eine Zufallszahl zwischen 0 und 0,99999999	RND(1)	RND(1)	RND(1)	RND(0)	RND	RND
Wähle eine Zufallszahl zwischen 1 und n	RND(N)	RND(1)*N+1	RND(1)*N+1	RND(N)	RND*N+1	RND*N+1
Wähle einen Zufallsbuchstaben zw. A und Z	CHR$(RND(26)+64)	CHR$(INT(RND(1) *26+65))	CHR$(INT(RND(1) *26+65))	CHR$(RND(26)+64)	CHR$(INT(RND*26+65))	CHR$(INT(RND*26+38))
Bildschirm räumen	CLS	PRINT CHR$(147)	HOME	CLS	CLS	CLS
Prüfe, ob auf der Tastatur eine Taste gedrückt wurde	INKEY$(N)	GET X$	X$=" " IF PEEK(–16384) >127 THEN GET X$	INKEY$	INKEY$	INKEY$
Wandle Zeichen in Codezahlen um	ASC("X") (benutzt ASCII-Code)	ASC("X") (benutzt ASCII-Code)	ASC("X") (benutzt ASCII-Code)	ASC("X") (benutzt ASCII-Code)	CODE("X") (benutzt ASCII-Code)	CODE("X") (benutzt ASCII-Code)
Cursor nach oben	PRINT CHR$(11)	PRINT CHR$(145)	CALL –998	PRINT CHR$(27)	PRINT CHR$(11)	PRINT CHR$(112)
Cusor nach unten	PRINT CHR$(10)	PRINT CHR$(17)	PRINT CHR$(10)	PRINT CHR$(26)	PRINT CHR$(10)	PRINT CHR$(113)
Cursor nach links	PRINT CHR$(8)	PRINT CHR$(157)	PRINT CHR$(8)	PRINT CHR$(24)	PRINT CHR$(8)	PRINT CHR$(114)
Cursor nach rechts	PRINT CHR$(9)	PRINT CHR$(29)	PRINT CHR$(21)	PRINT CHR$(25)	PRINT CHR$(9)	PRINT CHR$(115)
Nimm die ersten n Zeichen aus der Zeichenkette	LEFT$(A$,N)	LEFT$(A$,N)	LEFT$(A$,N)	LEFT$(A$,N)	A$(1 TO N)	A$(1 TO N)
Nimm die letzten n Zeichen aus der Zeichenkette	RIGHT$(A$,N)	RIGHT$(A$,N)	RIGHT$(A$,N)	RIGHT$(A$,N)	A$(N TO)	A$(N TO)
Nimm die mittleren n Zeichen aus der Zeichenkette	MID$(A$,N1,N2)	MID$(A$,N1,N2)	MID$(A$,N1,N2)	MID$(A$,N1,N2)	A$(N1 TO N2)	A$(N1 TO N2)

Dritter Teil

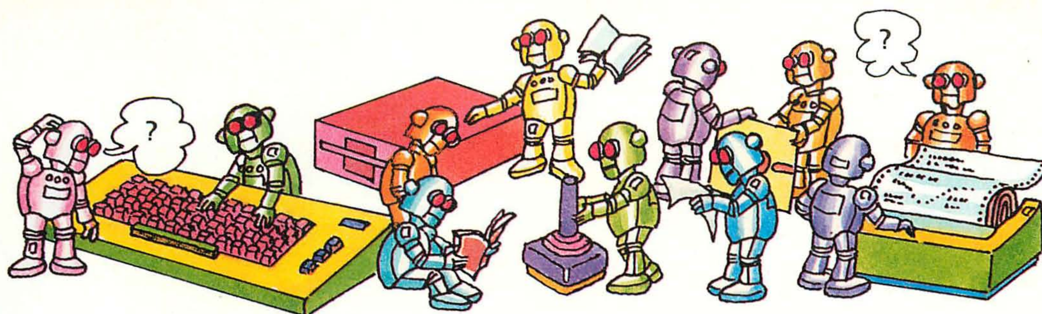
COMPUTER VON A-Z

Ein Bildwörterbuch

Von Corinne Stockley und Lisa Watts

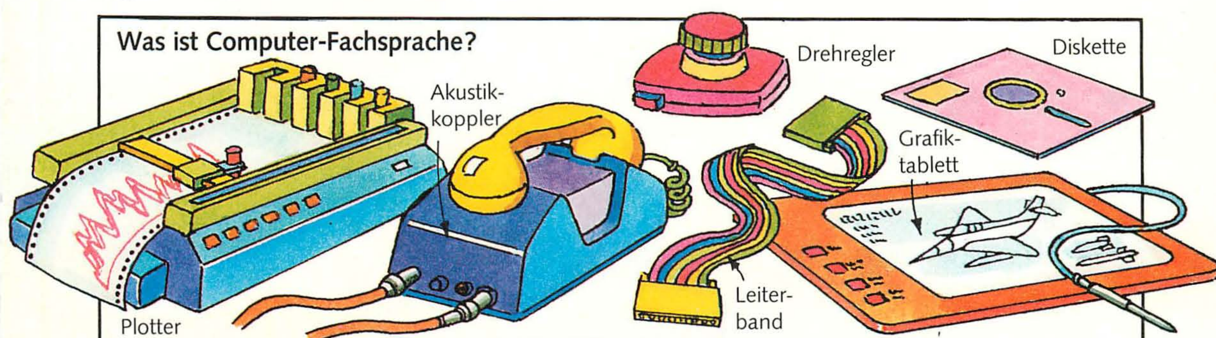


Computer von A – Z



Für viele Leute stellen die Begriffe, mit denen Computer beschrieben werden, eine ganz neue Sprache dar. Man braucht ein Wörterbuch, um eine Computer-Fachzeitschrift oder ein Computerbuch zu verstehen oder auch nur eine Werbeanzeige für Computer zu lesen. In diesem Teil des Buches werden die Fachbegriffe erläutert, die man kennen sollte, wenn man sich mit Handhabung und Funktion von Mikro- oder Heimcomputern vertraut machen möchte. Für jeden Begriff gibt es eine genaue Erklärung, oft sogar mit einer entsprechenden Illustration. Darüber hinaus wird auch auf Fachbegriffe eingegangen, die Sie zum Verständnis etwas speziellerer Texte benötigen.

Was ist Computer-Fachsprache?



Alle Fachgebiete haben ihre eigene Sprache, in der sich die Fachleute sehr genau untereinander verständigen können. So haben z. B. Mathematiker ihre mathematische Fachsprache, Musiker gebrauchen musikalische Begriffe, und Ingenieure benutzen technische Fachausdrücke. Computer beruhen auf einer so neuen technischen Entwicklung, daß man eine große Zahl von neuen Begriffen prägen mußte, um Dinge und Vorgänge zu beschreiben, die es vorher noch nicht gegeben hatte. Diese Begriffe haben die verschiedenartigsten Ursprünge. So gibt es z. B. Wörter aus unserer Umgangssprache, denen eine neue Bedeutung gegeben wurde. Es gibt technische Ausdrücke, es gibt Markennamen von Herstellern, und manchmal wurden aus Abkürzungen neue Wörter gebildet. Und ständig kommen neue Begriffe hinzu. Daher kann ein Buch über die Computer-Fachsprache kaum auf dem allerneuesten Stand sein. Sie finden hier jedoch einen Überblick über alle Begriffe, die sie brauchen, um sich ein grundlegendes Wissen über Computer zu verschaffen. Dabei werden Sie bald feststellen, daß das leichter ist, als Sie zunächst angenommen haben.

Wie man das Wörterbuch benutzt

Die Fachbegriffe in diesem Teil des Buches sind nach Sachgebieten geordnet. So sind z. B. alle Begriffe zusammengefaßt, die mit Programmen zu tun haben, ebenso die Begriffe, die von Grafik oder von der Speicherung handeln oder zur Beschreibung der Teile dienen, die sich im Inneren des Computers befinden. Eine Übersicht über die behandelten Themenbereiche finden Sie im Inhaltsverzeichnis auf der gegenüberliegenden Seite.
Wenn Sie die Erklärung für einen bestimmten Begriff suchen, so schlagen Sie diesen Begriff zunächst im Register am Ende des Buches nach. Die ausführliche Erklärung finden Sie auf der Seite, deren Seitenzahl **fett** gedruckt ist. Die übrigen Seitenzahlen geben an, wo dieser Begriff noch erscheint. Tauchen im fortlaufenden Text neue Fachbegriffe auf, so sind diese ebenfalls fett gedruckt. Das bedeutet, daß sie an anderer Stelle in diesem Teil des Buches erklärt werden; diese Stelle finden Sie mit Hilfe des Registers.

Der Computer

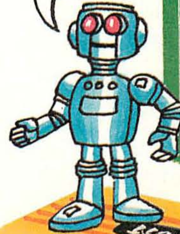
Ein **Computer** ist ein Gerät, das Informationen oder **Daten** verarbeitet. Es folgt dabei einer Folge von Befehlen, einem **Programm**. Informationen und Befehle sind abrufbereit im **Speicher** des Computers gespeichert und werden von der **Zentraleinheit** ausgeführt.

Hardware: Die Geräte, die zum Computer gehören, z. B. der **Bildschirm** und die **Tastatur**.

Software: Die Programme, die dem Computer sagen, was er zu tun hat, nennt man Software. Dies gilt sowohl für die Programme, die schon fest im Computer eingebaut sind, als auch für die Programme, die Sie eingeben.

Eingabe und **Ausgabe**: Die Informationen, die man einem Computer eingibt, heißen Eingabe; die Informationen, die der Computer ausgibt, heißen Ausgabe.

Tastatur: Mit Hilfe der Tastatur werden Informationen und Befehle in den Computer eingegeben. Die Tastatur sieht aus wie eine Schreibmaschinentastatur, sie hat jedoch meist zusätzliche Tasten für spezielle Befehle.



Verschiedene Arten von Computern

Mikrocomputer: Die Abbildung oben zeigt einen Mikrocomputer, oder kurz **Mikro**. Die meisten Mikrocomputer besitzen eine eingebaute Tastatur. Sie finden Verwendung in kleinen Betrieben, in Schulen und vor allem zu Hause. Man nennt sie auch **Personal Computer** oder **Heimcomputer**.

Großcomputer sind sehr große und leistungsstarke Computer, die riesige Mengen von Informationen verarbeiten und sehr viele Tätigkeiten auf einmal erledigen können. Mit allem notwendigen Zubehör brauchen sie viel Platz – oft mehrere Räume. Großcomputer werden vor allem in der Verwaltung eingesetzt, z. B. bei Versicherungen, Krankenkassen und Finanzämtern.

Minicomputer sind kleiner als Großcomputer. Auch sie können mehrere Dinge auf einmal erledigen, doch bei weitem nicht so viele wie ein Großcomputer.

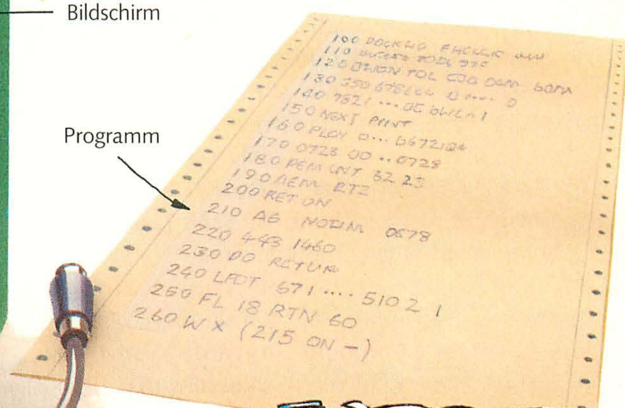
Taschencomputer sind so klein, daß sie in eine Tasche passen. Sie arbeiten mit Batteriestrom und haben normalerweise eine eingebaute Anzeige.

Spielcomputer: Diese Art von Computern wird nur für Spiele eingesetzt. Sie können normalerweise nicht programmiert werden, um andere Aufgaben zu erledigen.

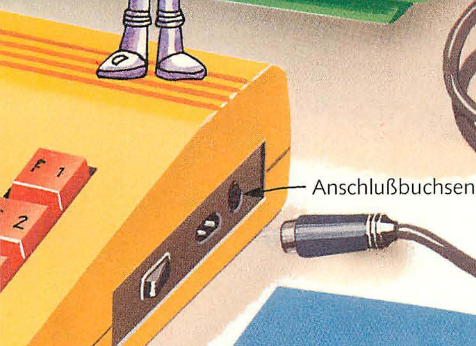
Weitere Informationen über **Programme** siehe Seite 128, über **Speicher** siehe Seite 106, über die **Zentraleinheit** siehe Seite 112.

Bildschirm: Die Informationen, die der Computer ausgibt, erscheinen auf dem Bildschirm. Der Bildschirm kann sowohl ein gewöhnlicher Fernsehbildschirm sein oder ein spezieller Bildschirm, den man **Monitor** nennt. Manche Computer haben eigene, fest eingebaute Bildschirme.

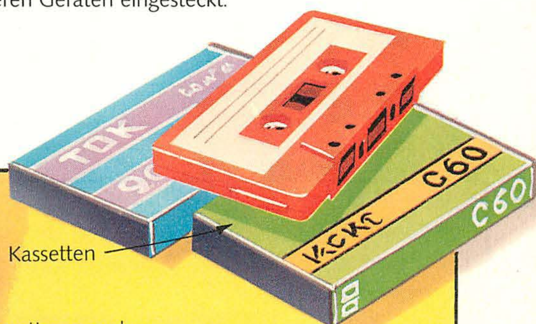








Anschlußbuchsen: In die Anschlußbuchsen des Computers werden die Stecker der Verbindungskabel zu anderen Geräten eingesteckt.





Zusatz- oder **Peripheriegeräte**: Jedes Gerät, das man mit einem Computer verbinden kann, ist ein Peripheriegerät. So kann man z. B. einen **Kassettenrecorder** oder ein **Diskettenlaufwerk** dazu verwenden, Programme und Informationen auf **Kassetten** oder **Disketten** zu speichern. Weitere Beispiele für Peripheriegeräte sind **Drucker**, die Programme oder Informationen auf Papier ausdrucken, oder **Plotter**, die Bilder zeichnen.

Die Tastatur

Die Tasten einer **Computertastatur** sind normalerweise angeordnet wie bei einer Schreibmaschine. Diese Tastatur heißt **QWERTZ-Tastatur**, entsprechend den ersten sechs Tasten der oberen Buchstabenreihe (bei englischen Geräten QWERTY). Die meisten Tastaturen sind **alphanumerisch**, d. h. sie haben sowohl Buchstaben- als auch Zifferntasten. Manche Tastaturen haben ein **numerisches** Tastenfeld; das ist ein Feld, in dem nur Zifferntasten vorkommen. Andere Tastaturen haben Tasten, die **Schlüsselwörter** enthalten. Die Befehlswörter stammen aus einer **Programmiersprache** (siehe Seite 126) und veranlassen den Computer, eine bestimmte Aufgabe auszuführen. Man braucht diese Tasten also nur einmal zu drücken, um dem Computer das ganze Wort einzugeben. Dies nennt man **Schlüsselworteingabe**.

Es gibt verschiedene Arten von Tastenfeldern: Die **Folientastatur** ist ganz flach, und die Tasten sind auf eine Spezialfolie gedruckt, die auf leichten Druck anspricht. Die **Taschenrechnertastatur** hat kleine, elastische Tasten. Die meisten Tastaturen haben bewegliche und voneinander getrennte Tasten wie auf einer Schreibmaschine. Die Wiederholung von Zeichen erfolgt bei manchen Tastaturen, indem man die entsprechende Taste gedrückt hält. Bei anderen Tastaturen gibt es eine **Wiederholungstaste**, die man nach der Taste mit dem Zeichen drückt, das wiederholt werden soll.

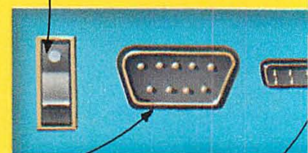


Mikrocomputer mit einer (englischen) QWERTY-Tastatur.

Umschalttaste: Auf einer Taste befinden sich normalerweise zwei oder mehrere Zeichen. Wenn man die Umschalttaste gleichzeitig mit einer beliebigen anderen Taste drückt, dann wird das Zeichen ausgegeben, das auf der Taste oben steht. Mit der **Feststelltaste** hält man die Umschalttaste fest, d. h. man schreibt z. B. nur Großbuchstaben.

Anschlüsse und Schalter

Ein-/Aus-Schalter: Damit schaltet man das Gerät ein oder aus.



Analoganschluß: Dient zum Anschluß von Zubehör wie **Steuerknüppel** oder **Lichtgriffel**.

Druckeranschluß: Dient zum Anschluß eines **Druckers.**

Weitere Tasten

Hier werden einige Sondertasten vorgestellt, die allerdings bei verschiedenen Gerätetypen unterschiedlich beschriftet sein können.

STOP BREAK ESCAPE

Diese Tasten bewirken die Unterbrechung eines Programms an beliebiger Stelle.

CLR HOME

Diese Taste bewirkt, daß der **Cursor** (die blinkende Positionsmarke) in die linke obere Bildschirmecke wandert. Drückt man diese Taste zusammen mit der **Umschalttaste**, dann wird der Bildschirm freigemacht.

RETURN NEWLINE ENTER

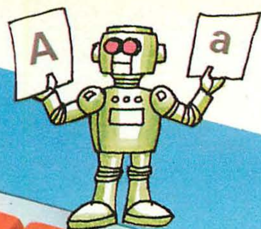
Diese Taste drückt man nach jedem Befehl und bewirkt dadurch, daß der Computer diesen Befehl ausführt oder speichert.

CTRL

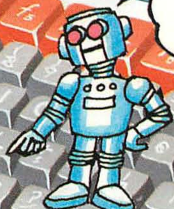
Dies ist eine **Steuertaste**. Drückt man die Taste CTRL zusammen mit einer anderen Taste, dann faßt der Computer dies als Steuerfunktion auf. Der Computer führt dann eine bestimmte Aufgabe aus, z. B. sendet er zum Drucker.

ESC

Diese Steuertaste benutzt man zusammen mit anderen Tasten, um diesen verschiedene Aufgaben zuzuweisen.



Der **Zeichensatz** ist die Menge der Zeichen, die ein Computer erzeugen kann. Die meisten Computer verwenden einen Zeichensatz, der dem **ASCII-Code** entspricht (ASCII = **A**merican **S**tandard **C**ode for **I**nformation **I**nterchange). Dieser Code enthält alle Buchstaben des Alphabets, die Ziffern 0 bis 9 und eine gewisse Anzahl von Symbolen.



Zeichen können Buchstaben, Zahlen oder Symbole sein.

Funktionstasten: Diese Tasten heißen auch **frei programmierbare Tasten**. Man kann den Computer programmieren, daß er bestimmte Aufgaben ausführt, indem man eine Funktionstaste drückt, z. B. für die Ausgabe eines Wortes wie PRINT oder INPUT. Diese Wörter braucht man sehr häufig, wenn man ein Programm schreibt. Beim ZX 81 verhält sich die Funktionstaste wie eine zweite **Umschalttaste**.

Leertaste: Damit kann man zwischen einzelnen Wörtern oder Buchstaben Leerstellen lassen.

Mit den **Cursor-Steuertasten** bewegt man den Cursor nach oben, unten, nach links oder rechts. Der Cursor ist das Zeichen, das anzeigt, an welcher Stelle des Bildschirms das nächste Zeichen erscheinen wird.

Cass: Diese Buchse dient zum Anschluß eines **Kassettenrecorders**. Manchmal gibt es dafür zwei verschiedene Anschlüsse. Einer dient dazu, die Signale vom Kassettenrecorder zu empfangen (**EAR**), der andere dient dazu, die Signale zum Recorder zu senden (**MIC**).



Video: Diese Buchse dient zur Verbindung des Mikrocomputers mit dem **Monitor** (siehe Seite 104).

UHF: Buchse zur Verbindung mit dem Fernsehgerät. UHF bedeutet **U**ltra**h**och**f**requenz. Diese Signale werden auch beim Fernsehen verwendet.

RESET: Bewirkt, daß der Computer aufhört zu arbeiten.

User Port: Über diesen Anschluß läßt sich der Computer mit anderen Geräten verbinden, die man steuern möchte, z. B. mit einer Modelleisenbahn oder einem Roboter.


Multiplikationszeichen

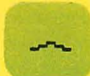

Divisionszeichen


Zeichen für „größer als"


Zeichen für „kleiner als"


Dollar- oder **String**-Zeichen. Wird für bestimmte Arten von **Variablen** verwendet (siehe Seite 129).


Zeichen für „null"

▲ Alle diese Zeichen bedeuten „Potenzieren".


Mit diesem Zeichen werden **Hexadezimal**zahlen bezeichnet, z. B. #6.


Zeichen für „und" oder zur Kennzeichnung von Hexadezimalzahlen.

▲ Alle diese Zeichen stammen aus dem **ASCII-Code**. Die meisten von ihnen haben keine besondere Bedeutung, man kann sie jedoch auf dem Bildschirm ausdrucken, genau wie jedes andere Zeichen.

Der Bildschirm

Alles, was man einem Computer eingibt, und alle Ergebnisse, die der Computer errechnet, können auf dem **Bildschirm** sichtbar gemacht werden. Dieser Bildschirm kann ein normales Fernsehgerät sein oder ein Monitor. Er sieht genauso aus wie ein Fernsehgerät, man kann damit aber keine Fernsehprogramme empfangen.



Dieser Bildschirm zeigt Informationen, die vom Computer kommen.

Cursor: Dies ist ein Symbol, das auf dem Bildschirm anzeigt, an welcher Stelle das nächste Zeichen erscheinen wird. Je nach Gerät zeigt sich der Cursor als blinkendes Quadrat, als Pfeilspitze, als K oder als kleiner Strich.

Bereitzeichen: Dieses Zeichen – meist ein Fragezeichen – erscheint auf dem Bildschirm, sobald der Computer Informationen erwartet.

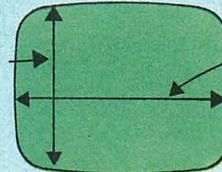
Überlauf: Sobald der Bildschirm mit Text „vollgeschrieben" ist, werden automatisch alle Zeilen nach oben bewegt, so daß am unteren Bildschirmrand neuer Freiraum erscheint.

Negativschrift: Dies sind Zeichen, die z. B. aus hellen Buchstaben auf dunklem Untergrund bestehen können, wie in der Abbildung oben.

Zeilenübergang: Der Übergangsprozeß läuft automatisch ab, indem der Computer eine neue Zeile beginnt, sobald eine Zeile vollgeschrieben ist.

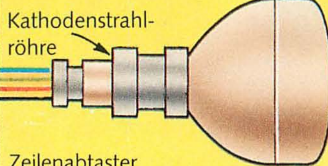
Vordergrund und Hintergrund: Manchmal auch **Tinte und Papier** genannt. Bezieht sich auf die Farbe, in der der Text ausgedruckt wird bzw. auf die Farbe des Bildschirmuntergrunds.

Bildschirmgröße: Darunter versteht man die Anzahl der **Zeichen**, die auf den Bildschirm passen. Sie unterscheidet sich von Computer zu Computer. Die Breite



des Bildschirms entspricht der Anzahl der Zeichen, die in einer Bildschirmzeile Platz haben. Die Höhe des Bildschirms entspricht der Anzahl der Zeilen, die übereinander stehen können.

Bildschirmarten



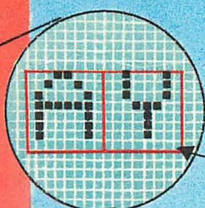
Kathodenstrahlröhre: Diese Röhre im Inneren eines Fernsehgeräts erzeugt den Elektronenstrahl, der das Bild auf dem Bildschirm erscheinen läßt.

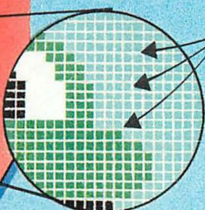
Zeilenabtastung: Das Bild auf dem Bildschirm entsteht auf folgende Weise: Der Elektronenstrahl bewegt sich auf dem Bildschirm so schnell von oben links nach unten rechts, daß man es nicht sehen kann. Dabei wird das Bild ständig wiederholt.

PAL: In den meisten Ländern Westeuropas (außer Frankreich) ist dies der Standard für Fernsehbilder. Dieser Standard gibt an, wie viele Zeilen der Elektronenstrahl auf dem Bildschirm durchläuft und mit welcher Geschwindigkeit dies geschieht.

NTSC: Dies ist der Fernsehstandard in den USA; er gibt an, wie viele Zeilen auf dem Bildschirm geschrieben werden und wie schnell dies geschieht.

Zeichenmatrix: Jedes Zeichen auf dem Bildschirm entsteht dadurch, daß innerhalb eines Punktrasters einzelne Punkte aufgehellt werden. Gewöhnlich umfaßt ein solches Rasterfeld 8 mal 8 bzw. 7 mal 8 Punkte. Dieses Feld heißt Zeichenmatrix.





Bildpunkte: Darunter versteht man die Punkte, die der Computer aufleuchten läßt; auf diese Weise werden die Bilder auf dem Bildschirm erzeugt.

Anzeige-Betriebsarten

Der Begriff **Betriebsart** bezieht sich auf einen gewissen Operationszustand eines Computers, in dem er unterschiedliche Befehle verstehen kann. Die meisten Computer haben verschiedene **Anzeige-Arten**, sind daher also in der Lage, verschiedene Anzeigen auf den Bildschirm zu bringen. Bei einem **Text-Betrieb** kann man beliebigen Text auf dem Bildschirm erzeugen, z. B. Datenzeilen und Programmzeilen. Manche Computer haben verschiedene Text-Betriebsarten, mit denen sie z. B. Texte mit unterschiedlicher Zeichenanzahl ausdrucken können, je nach Breite und Höhe des Bildschirms.

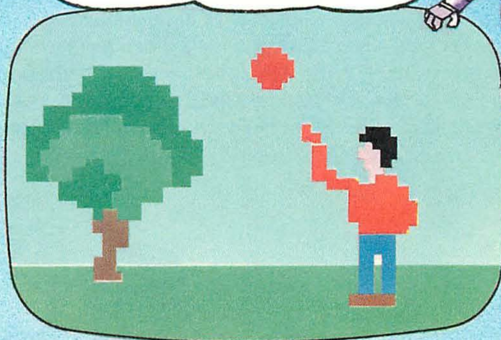
Teletext-Betrieb: Diese Betriebsart erlaubt dem Computer, die notwendigen grafischen und sonstigen Zeichen anzuzeigen, die mit Hilfe von Fernsehsignalen ausgesendet werden.

Auflösung: Sie ist gekennzeichnet durch die Anzahl der Bildpunkte, die der Computer auf dem Bildschirm aufleuchten lassen kann. Sie wird auch **Darstellungsgenauigkeit** genannt.



Hohe Auflösung: Bei einem Bildschirm mit hoher Auflösung kann der Computer eine große Anzahl von sehr kleinen Bildpunkten aufleuchten lassen und dadurch sehr scharfe Bilder erzeugen.

Mehr Informationen über **Bildpunkte** und **Grafik** auf Seite 124/125.



Geringe Auflösung: Bei geringer Auflösung sind die Bilder, die der Computer erzeugt, nicht sehr scharf, da die einzelnen Bildpunkte ziemlich groß sind und außerdem weniger Bildpunkte als bei hoher Auflösung vorhanden sind.



LCD oder **Flüssigkristall-Anzeige**: Das Bild auf dem Bildschirm wird durch einen Flüssigkristall erzeugt, der sich dunkel färbt, sobald eine elektrische Spannung gelegt wird. Manche kleinen Computer besitzen Flüssigkristall-Bildschirme.

Monochrom: Dies ist ein einfarbiger oder Schwarzweiß-Bildschirm. Die Farbe ist gewöhnlich grün oder bernsteingelb.

RGB-Monitor: Dies ist ein besonderer Farb**monitor**. Seine Anzeige entsteht durch drei verschiedene Signale, je eines für **R**ot, **G**rün und **B**lau (RGB). Aus diesen Farben wird das Bild auf dem Bildschirm zusammengesetzt. Es ist klarer und schärfer als das Bild, das bei einem Farbfernsehgerät durch ein kombiniertes Signal erzeugt wird.

Luminosität: Darunter versteht man die Helligkeit eines Bildschirms. Manche Bildschirme haben eine flimmerfreie Oberfläche, damit die Augen nicht ermüden, wenn man ihn längere Zeit beobachten muß.

Ein **kompatibler Bildschirm für Video- und Bildschirmtext** hat das geeignete Format für die Anzeige beider Systeme.

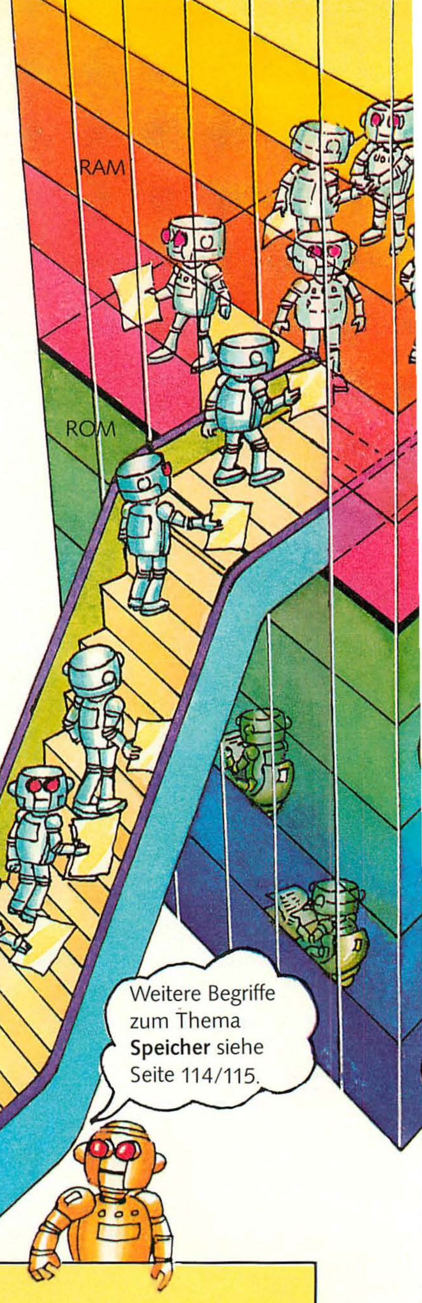
Speicher

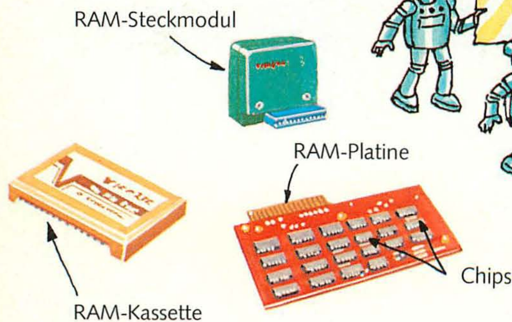
Der **Speicher** ist ein Bereich im Computerinneren, in dem Informationen und Befehle gespeichert werden. Bei diesem Speicher spricht man daher von einem **internen Speicher**. Unter **externen Speichern** versteht man Programme und Daten, die auf **Kassetten** oder **Disketten** gespeichert werden (siehe Seite 118/119) und bei Bedarf in den internen Speicher des Computers geladen werden können. Der Speicher des Computers besteht aus zwei Bereichen: **RAM** und **ROM**.

RAM ist die Abkürzung für **R**andom **A**ccess **M**emory (soviel wie Direktzugriffspeicher). In diesem Speicher werden alle Informationen gespeichert, die man in den Computer eingibt. Man kann auch jederzeit Informationen eingeben und abrufen (daher auch **Schreib-/Lesespeicher** genannt).

Das RAM ist meist ein flüchtiger Speicher, d. h. die im RAM gespeicherte Information ist verloren, sobald der Computer ausgeschaltet wird. Man nennt ihn daher auch **zeitweiligen Speicher**. Er kann sowohl statisch als auch dynamisch sein: Ein **statischer RAM-Speicher** empfängt lediglich dann einen elektrischen Impuls, wenn sein Inhalt geändert werden muß. Ein **dynamischer RAM-Speicher** benötigt ständig elektrische Impulse, um seine Informationen zu behalten.

ROM ist die Abkürzung für **R**ead **O**nly **M**emory (soviel wie Nur-Lese-Speicher). In diesem Speicher ist das Betriebssystem des Computers gespeichert. Neue Informationen können hier nicht aufgenommen werden. Das ROM ist ständig vorhanden, d. h. die dort gespeicherten Informationen sind ständig verfügbar, selbst wenn der Computer ausgeschaltet ist. Das ROM heißt daher auch **Festwertspeicher**.



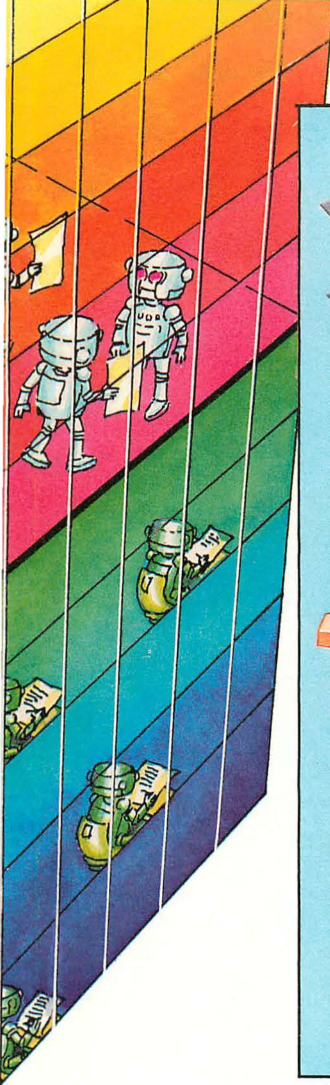


Speicherplatz-Erweiterung

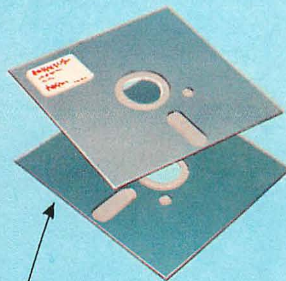
Wenn man den Speicherplatz eines Computers erweitern möchte, um längere Programme verwenden und mehr Informationen speichern zu können, muß man den RAM-Speicher vergrößern. Die Erweiterung des Speichers erfolgt mittels eines **RAM-Steckmoduls** oder einer **Kassette**, die in den Erweiterungsanschluß des Computers gesteckt werden. Man kann auch eine RAM-**Platine** an dem dafür vorgesehenen Platz auf der Hauptplatine im Inneren des Computers einstecken. Man spricht dann von zusätzlichem RAM-Speicher.

Die Speicherkapazität

Die Information gelangt als Folge von elektrischen Signalen in den Speicher. Es gibt zwei Arten von Signalen – „Impuls" und „kein Impuls". Diese Signale heißen **Bits**, und sämtliche Informationen, die ein Computer erhält, werden in dieser Form eingegeben. Jeder Buchstabe, jede Zahl bzw. jedes Symbol wird von einer Gruppe von acht Bits dargestellt. Eine solche Achtergruppe heißt **Byte**.
Das Fassungsvermögen des Speichers wird in **Kilobytes** (KB) angegeben; ein Kilobyte sind 1 024 Bytes. Dies bietet genügend Speicherplatz für über zweihundert Wörter. Ein **Megabyte** sind eine Million Bytes.



Externe Speicher



Disketten

Programme oder Daten, die im RAM des Computers gespeichert sind, gehen normalerweise verloren, sobald der Computer ausgeschaltet wird. Um sie zu erhalten, ist es erforderlich, diese Programme oder Daten auf einer **Kassette** oder **Diskette** (**Floppy disk**) zu speichern. Man kann auch Kassetten und Disketten mit gespeicherten Programmen kaufen. Auf diese Weise gespeicherte Informationen sind externe Speicher.



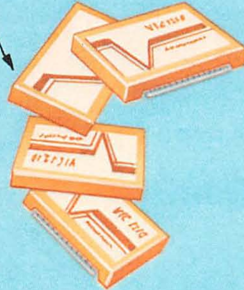
Kassetten

SAVE bedeutet, daß man Programme oder Daten auf Kassette oder Diskette speichert („sichert").

LOAD bedeutet, daß man Programme oder Daten von einer Kassette oder Diskette in den Speicher des Computers lädt.



ROM-Kassetten



Zugriffszeit nennt man die Zeit, die der Computer benötigt, um Programme oder Daten auf einer Kassette oder Diskette zu finden.

ROM-Kassetten enthalten Programmkopien. Man kann sie in die **Tastatur** (entweder hinten oder an der Seite) einstecken. Diese Kassetten enthalten z. B. Spielprogramme oder Anweisungen, die dem Computer sagen, wie er **Grafik** oder Musik erzeugen kann.

Mikro-Bandlaufwerk: Dies ist eine Kassette, die eine Magnetbandschleife enthält, auf der man Programme oder Daten speichern kann. Sie arbeitet fast so schnell wie eine **Diskette** (siehe Seite 118), erfordert aber den Einsatz eines speziellen Geräts.

Kassetten

Die meisten Kassetten, auf denen man Programme oder Daten speichert, sind übliche Tonbandkassetten – die gleichen, die man auch für die Aufnahme von Musik benutzt. Man kann auch spezielle **Datenkassetten** kaufen, die für die Aufnahme von Computersignalen geeignet sind.
Eine Kassette enthält ein Magnetband, auf dem die Signale als Folge unterschiedlicher magnetischer Stärke aufgezeichnet werden. Wenn man eine Kassette abhört, auf der ein Computerprogramm gespeichert ist, kann man die Signale als Pieptöne hören.

Bandzähler: Die meisten Kassettenrecorder haben einen eingebauten Bandzähler. Er zählt von 0 bis 9 999, während das Band läuft. Beim Start des Programms setzt man den Bandzähler auf 0 und kann dann notieren, wo die einzelnen Programme beginnen.



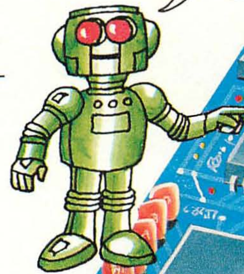
Kassette

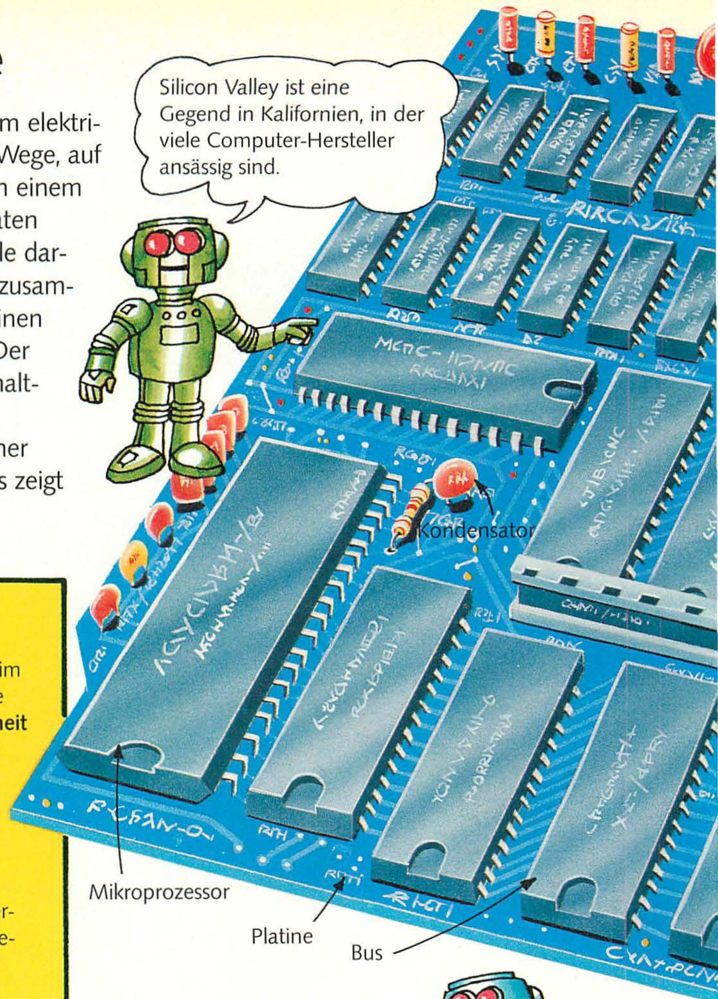
Vorspannband: Dies ist ein meist farbiges Plastikband am Anfang des Kassettenbands. Auf das Vorspannband kann man nichts aufnehmen.

Kopfstück: Es steht vor einem Programm oder einem Datenblock, der von einer Kassette geladen wird, und gibt dem Computer Informationen über das, was geladen wurde, z. B. über die Länge eines Datenblocks.

Das Tastaturgehäuse

Ein Computer ist ein kompliziertes System elektrischer **Schaltkreise**. Schaltkreise sind die Wege, auf denen elektrischer Strom fließen kann. In einem Computer werden auf diesen Wegen Daten transportiert, die durch elektrische Signale dargestellt werden. Eine bestimmte Anzahl zusammengehöriger Schaltkreise nennt man einen **integrierten Schaltkreis (IC)** oder **Chip**. Der Name „Chip" stammt daher, daß die Schaltkreise sich auf kleinen Siliziumplättchen befinden. Computerchips werden auf einer **Platine** angebracht. Die Abbildung rechts zeigt die Platine eines Mikrocomputers.





Verschiedene Arten von Chips

Mikroprozessor: Dies ist der wichtigste Chip im Inneren eines Mikrocomputers. Er enthält die Schaltkreise, die die Funktion der **Zentraleinheit** erfüllen. Die Zentraleinheit führt die gesamte Arbeit im Computer aus, d. h. sie steuert die Ausführung von Anweisungen und den Informationsfluß.

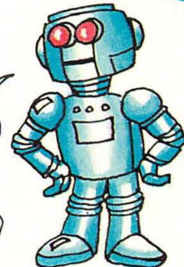
Speicherchips: In diesen Chips werden Programme und Daten gespeichert. Da es verschiedene **Speicher** gibt, gibt es auch verschiedene Speicherchips (siehe unten).

ROM-Chips: Diese **Festwertspeicher-Chips** enthalten die Programme, mit deren Hilfe der Computer alle Aufgaben erledigt, die ihm gestellt werden.

RAM-Chips: In diesem **Schreib-/Lesespeicher-Chip** werden die Programme und Daten gespeichert, die man dem Computer eingibt.

Eingabe-/Ausgabe-Chip: Diese Chips steuern den Informationsfluß zwischen dem Computer und anderen Geräten, z. B. dem **Bildschirm**, der **Tastatur** oder dem **Drucker**. Eingabe-/Ausgabe-Chips werden in **Schnittstellen** verwendet (siehe Seite 111).

Dedizierte Chips: Dies sind zweckgebundene Chips, die nur eine Aufgabe ausführen können. Ein Speicherchip ist z. B. ein dedizierter Chip.

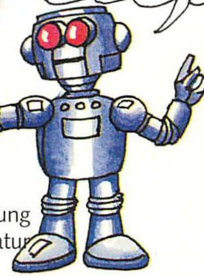


Platinen und Busse

Platine: Die Platine ist ein Kunststoffbrett mit dünnen Kupferbahnen, die den elektrischen Strom leiten.

Platinenanschluß (auch Edge connector): An dieser Stelle reichen die Metallbahnen bis an den Rand der Platine, so daß man Zusatzspeicher oder Programmkassetten dort anschließen kann.

Bus: Darunter versteht man ein Bündel von Metallbahnen auf einer **Platine**. Sie dienen dazu, die Computersignale zwischen den einzelnen Chips zu transportieren. Da verschiedene Signale transportiert werden müssen, gibt es auch verschiedene Busse (siehe Seite 110).



Modulator: Dieses auch **UHF-Modulator** genannte Gerät wandelt Computersignale in Signale um, die ein Fernsehgerät verarbeiten kann. UHF bedeutet **U**ltra**h**och**f**requenz; auch das Fernsehen verwendet UHF-Signale. **Monitoren** (siehe Seite 104) verwenden die gleichen Signale wie Computer und brauchen daher keinen Modulator.

Spannungsregler: Dieser Regler kontrolliert die vom Computer benötigte 5-Volt-Spannung.

Bus-System: Darunter versteht man die Anordnung der Leiterbahnen auf der Platine zum Austausch der Signale zwischen den einzelnen Bauteilen. Ein bekannter Systembus für Mikrocomputer ist der **S-100**.

Mutterplatine: Auf diese Platine können **Tochterplatinen** aufgesteckt werden. Die Mutterplatine wird dann seitlich oder an der Rückseite des Computers angebracht. Manchmal wird auch die Hauptplatine im Inneren des Computers so bezeichnet.

Karte: Darunter versteht man eine kleine Platine, z. B. eine Tochterplatine, die man auf die Hauptplatine aufstecken kann. Auf einer Karte befinden sich z. B. weitere **Speicherchips** oder **Schnittstellen**-Chips.

Elektronische Bausteine

Darunter versteht man kleine Geräte, die den Fluß des elektrischen Stroms auf der **Platine** steuern. Es gibt verschiedene Arten von elektronischen Bausteinen.

Transistor: Er verhält sich wie ein Schalter und kann den elektrischen Strom anhalten oder durchlassen. Transistoren steuern den Fluß der Signale („Impuls" oder „kein Impuls"), mit denen der Computer arbeitet.

Diode: Eine Diode bewirkt, daß elektrische Signale nur in einer Richtung fließen.

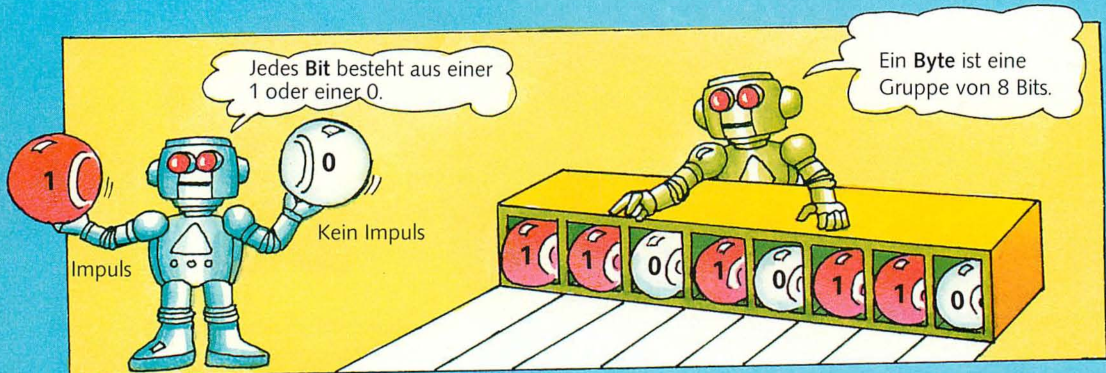
Leuchtdiode (LED): Diese Diode sendet Licht aus, da sie aus einem Material hergestellt ist, das leuchtet, sobald elektrischer Strom hindurchläuft. Leuchtdioden werden oft zur Anzeige eingesetzt, ob das Gerät ein- oder ausgeschaltet ist.

Widerstand: Er steuert die Stärke des elektrischen Stroms.

Kondensator: Er kann kleine Strommengen speichern und wird dazu verwendet, den Elektrizitätsfluß im Inneren des Computers zu steuern und zu regulieren.

Bits und Bytes

Der Computer verarbeitet Informationen in Form von Binärzahlen. Der **Binärcode** ist ein Zahlensystem, in dem alle Zahlen nur mit den beiden Ziffern 1 und 0 dargestellt werden (siehe Seite 42).



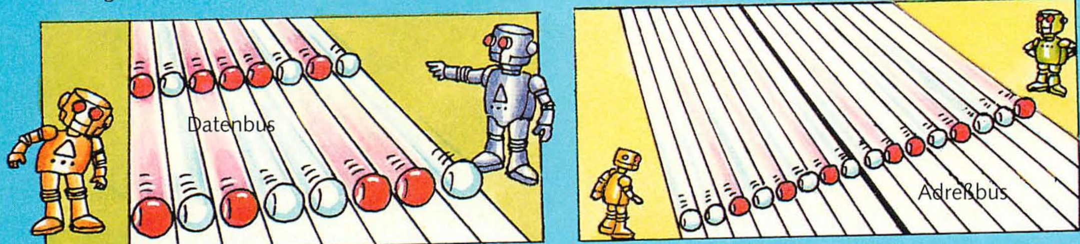
Die einzelnen Ziffern heißen **Bits** (Kurzform für **bin**ary dig**its**) und werden dargestellt durch einen elektrischen Impuls für 1 und keinen Impuls für 0. Die gesamte Arbeit im Inneren des Computers erfolgt durch die Verwendung von solchen Impuls- und Kein-Impuls-Ketten.

Daten werden dargestellt durch 8-Bit-Gruppen, die **Bytes** genannt werden. Die Bit-Menge, die ein Computer auf einmal verarbeiten kann, nennt man auch **Wort**. Ein 8-Bit-Computer arbeitet also mit 8-Bit-Wörtern, während ein 16-Bit-Computer Operationen mit 16-Bit-Wörten ausführt.

ASCII-Code: Dieser Code ist ein Standardcode für alle **Zeichen**, die ein Computer benutzt. Jeder Buchstabe, jede Ziffer und jedes Symbol werden in der Regel von einem Byte dargestellt, das aus acht Bits besteht, obwohl tatsächlich nur sieben Bits davon den ASCII-Code bilden; das restliche Bit heißt **Paritäts-Prüfbit**. Der Computer prüft damit, ob sich die Bits auf ihrem Weg von einem Teil des Computers zu einem anderen nicht versehentlich geändert haben. Ist eine ungerade Anzahl von Einsen in den sieben Bits enthalten, dann wird das Prüfbit auf 1 gesetzt. Dadurch wird die Gesamtzahl der Einsen eine gerade Zahl. Dies nennt man **gerade Parität**. Man kann auch eine **ungerade Parität** erzielen; in diesem Fall wird das Bit so bestimmt, daß die Gesamtzahl der Einsen eine ungerade Zahl ergibt. Das Byte wird beim Empfang mit dem Prüfbit verglichen.

Busse

Man kann sich vorstellen, daß die acht Bits, die ein Byte ausmachen, sich auf acht Bahnen parallel zueinander auf der **Platine** fortbewegen. Eine solche Bahngruppe nennt man Bus. Es gibt verschiedene Busse für verschiedene Aufgaben.



Datenbus: Dieser Bus transportiert Byte für Byte (oder Wort für Wort) Daten von der **Zentraleinheit** zu einem **Speicher** oder einem **Peripheriegerät**, z. B. zu einem Drucker.

Steuerbus: Dieser Bus transportiert Signale von der Zentraleinheit zur Steuerung verschiedener Tätigkeiten. So wird z. B. ein Signal ausgesendet, das dem Speicher sagt, ob Information gespeichert oder ausgegeben werden soll.

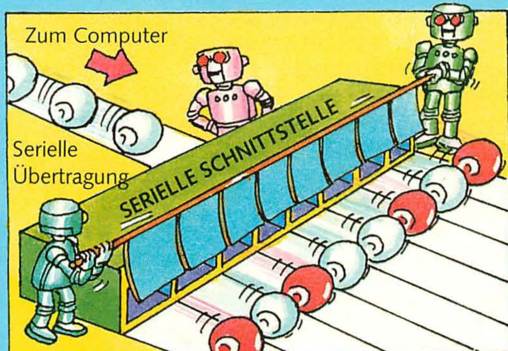
Adreßbus: Dieser Bus transportiert **Adressen**. Darunter versteht man die Zahlen, die einen bestimmten Speicherplatz im Speicher des Computers bezeichnen. Der Adreßbus besteht bei einem 8-Bit-Computer (siehe oben) aus 16 parallelen Bahnen, auf denen 16 Bits parallel in zwei Bytes transportiert werden.

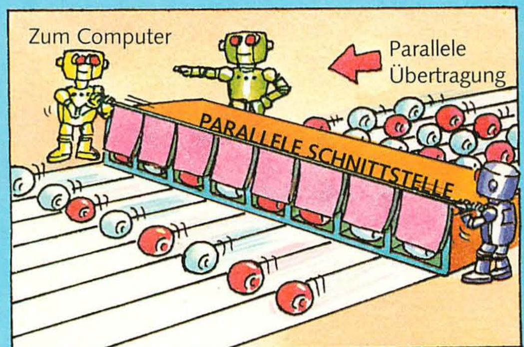
Schnittstellen

Die **Schnittstelle** ist eine Schaltung zur Verarbeitung von Signalen, die zwischen dem Computer und einem **Peripheriegerät**, z. B. einem Kassettenrecorder oder einem Drucker, hin- und herwandern. Die Schnittstelle wandelt Computersignale in solche Signale um, die das andere Peripheriegerät „versteht", und umgekehrt. Schnittstellen für normale Peripheriegeräte, z. B. für Fernsehgeräte oder Kassettenrecorder, sind normalerweise schon in den Computer eingebaut. Die Schnittstelle steuert auch die Geschwindigkeit, mit der Daten zwischen dem Computer und dem Peripheriegerät hin- und hertransportiert werden. Diese Geschwindigkeit wird in **Bits** pro Sekunde (Bps) gemessen und heißt **Baud**. Ein **Kilobaud** sind 1 000 Bits pro Sekunde. Werden Daten **synchron** übermittelt, dann sind die Zeitintervalle zwischen den Signalen regelmäßig. Bei **asynchroner** Übermittlung sind sie unregelmäßig. In diesem Fall werden besondere Bits, **Startbits** und **Stoppbits**, ausgesendet; sie geben an, wann ein Byte zu Ende ist und das nächste beginnt.

Verschiedene Arten von Schnittstellen

Es gibt im wesentlichen zwei Arten von Schnittstellen – **parallele Schnittstellen** und **serielle Schnittstellen**. Im Inneren des Computers werden alle Signale parallel transportiert. Daher wird oft auch eine parallele Schnittstelle verwendet, um den Computer mit einem Peripheriegerät zu verbinden, das gleichfalls parallel arbeitet. Eine serielle Schnittstelle wandelt die parallelen Signale des Computers in solche um, die hintereinander, also „in Serie", transportiert werden, und umgekehrt. Durch serielle Schnittstellen kann man Computer am einfachsten mit Peripheriegeräten verbinden, die in Serie arbeiten. Es gibt verschiedene Standardschnittstellen (siehe unten), die auf der Grundlage international vereinbarter Spannungs- und Strombedingungen arbeiten.



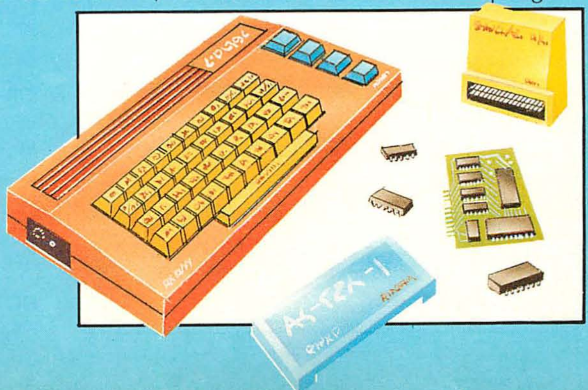


Standard-Schnittstellen

RS-232 C: Dies ist die am häufigsten verwendete Serien-Schnittstelle. Man bezeichnet sie auch als **V24**.
RS-423: Dies ist eine Serien-Schnittstelle, eine neuere Version der RS-232-Schnittstelle.

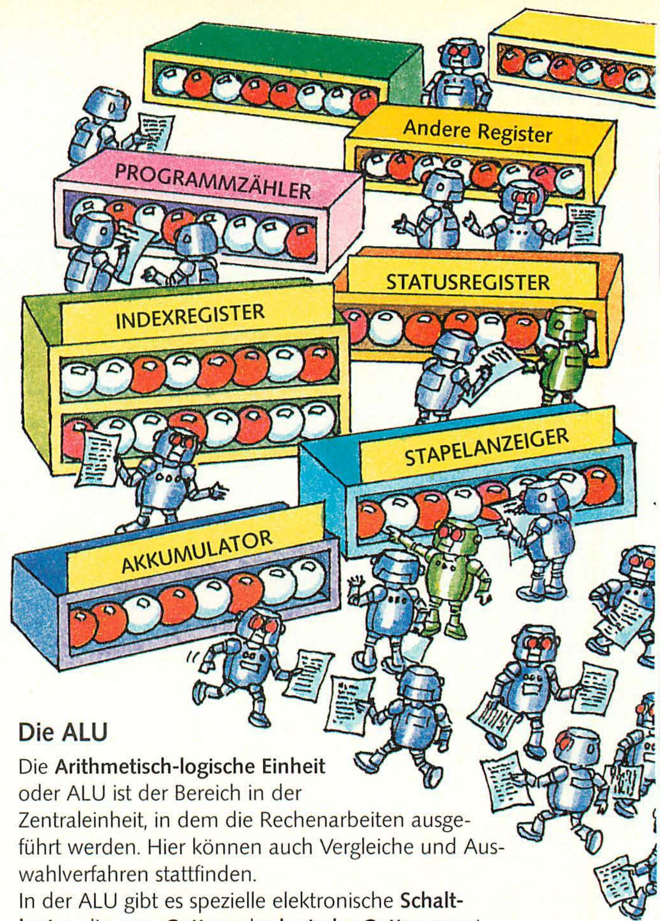
Centronics: Dies ist eine verbreitete parallele Schnittstelle, vor allem zum Anschluß von **Druckern**.
IEEE-488: Dies ist eine parallele Standard-Schnittstelle.

Quittungsbetrieb: Dieser Ausdruck beschreibt den Austausch von Nachrichten zwischen einem Computer und einem Peripheriegerät und drückt aus, daß diese Geräte bereit sind, Nachrichten zu senden bzw. zu empfangen.
Aufrüsten: Man rüstet einen Computer auf, indem man seine Fähigkeiten vergrößert. Dies kann geschehen, indem man weitere **Speicherchips** oder weitere Schnittstellen anschließt, um den Computer mit einer größeren Zahl von Peripheriegeräten zu verbinden. Die meisten Computer werden in zwei Versionen hergestellt: in einer Grundversion und einer erweiterten Version, die mehr Speicher und Schnittstellen bietet. Normalerweise kann man die Grundversion zur erweiterten Version aufrüsten.



Die Zentraleinheit

Die **Zentraleinheit** ist in einem einzelnen Chip enthalten, den man **Mikroprozessor** nennt. Sie ist der Teil im Computer, in dem alle Rechen- und Steuerarbeiten ausgeführt werden, die zur Erledigung bestimmter Aufgaben erforderlich sind, und besteht aus drei Hauptteilen: aus einer Anzahl von **Registern**, der **Arithmetisch-logischen Einheit (ALU)** und einer **Steuereinheit.**



Die Register

Die Register sind spezielle Speicherplätze, in denen Daten gespeichert werden, während die Zentraleinheit damit arbeitet. Die meisten Register können einen oder zwei **Bytes** speichern und werden gewöhnlich als „Lagerplätze" benutzt, sobald Daten zwischen Zentraleinheit und **Speicher** transportiert werden. Einige Register haben eine spezielle Aufgabe:

Akkumulator: Dieses Register wird von der Arithmetisch-logischen Einheit benutzt, um Zahlen zu speichern, die in Rechnungen verwendet werden.

Programmzähler: Er hält die **Adresse** an der Stelle im Speicher, wo die Anweisung gespeichert ist, die als nächste ausgeführt werden soll.

Statusregister: Wird auch **Bedingungscode-Register** genannt. Jedes **Bit** in diesem Register ist ein sogenanntes **Kennzeichenbit**. Es registriert, ob eine bestimmte Bedingung erfüllt ist oder nicht. So heißt z. B. eines der Bits **Übertragskennzeichen** oder **Carry Flag**. Es wird auf 1 gesetzt, wenn das Ergebnis der letzten Rechnung zu groß ist, um in der jeweiligen Wortlänge Platz zu finden.

Indexregister: Die Inhalte dieses Registers werden verwendet, wenn die Adresse der Information erarbeitet wird, die als nächste benutzt werden soll (siehe **indiziertes Adressieren**, Seite 135).

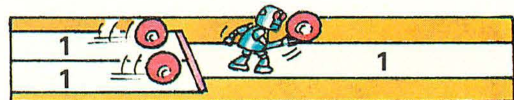
Stapelanzeiger: Er enthält die Adresse des nächstmöglichen freien Platzes in einem **Stapel**. Dies ist ein spezieller Platz im Speicher.

Hausarbeit: So nennt man die Aufgaben, die die Zentraleinheit ausführt und die nicht direkt in Verbindung mit der Lösung eines Programmproblems stehen. Damit können z. B. Eingabe und Ausgabe von Daten gemeint sein.

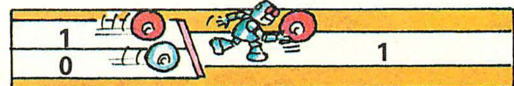
Treiber: Dieses Programm versetzt die Zentraleinheit in die Lage, **Peripheriegeräte** zu steuern, d. h. eine Kassette durchzusehen oder einen Drucker zu benutzen.

Die ALU

Die **Arithmetisch-logische Einheit** oder ALU ist der Bereich in der Zentraleinheit, in dem die Rechenarbeiten ausgeführt werden. Hier können auch Vergleiche und Auswahlverfahren stattfinden.
In der ALU gibt es spezielle elektronische **Schaltkreise**, die man **Gatter** oder **logische Gatter** nennt. Sie bestehen aus **Transistoren**, die die elektrischen Impulse, mit denen der Computer arbeitet, aussenden oder anhalten können. Es gibt verschiedene Arten von Gattern. Sie werden in verschiedenen Folgen zusammengestellt und arbeiten im **Binärcode** (siehe Seite 138). Die drei Haupttypen sind das **AND-Gatter**, das **Or-Gatter** und das **NOT-Gatter.**



AND-Gatter: Sendet einen elektrischen Impuls (eine „1") aus, sobald es an beiden Eingängen einen Impuls empfängt.



OR-Gatter: Sendet einen Impuls aus, wenn es an einem der beiden oder an beiden Eingängen einen Impuls empfängt.



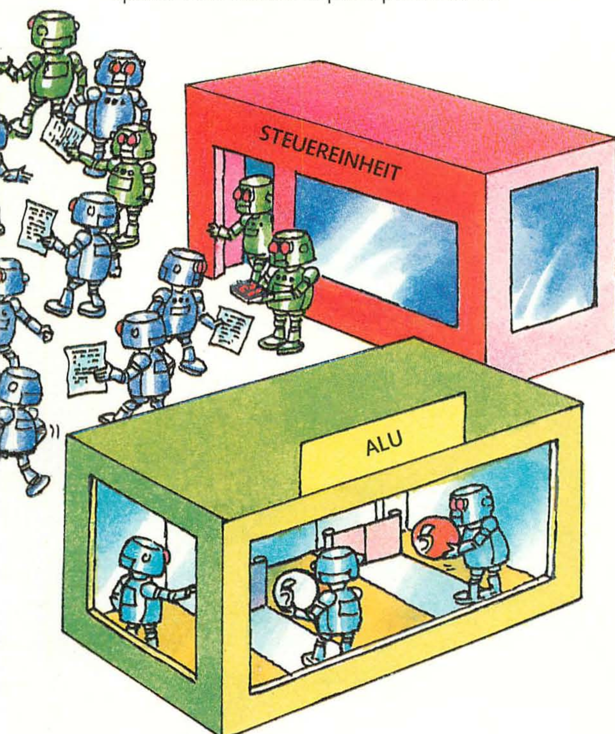
NOT-Gatter: Sendet nur dann einen Impuls aus, wenn es an seinem Eingang keinen Impuls empfängt.



Die Steuereinheit

Die **Steuereinheit** ist der Teil der **Zentraleinheit**, die alle Vorgänge (**Operationen**), die zur Erledigung einer Aufgabe erforderlich sind, steuert und koordiniert; sie holt z. B. Daten aus dem **Speicher** und schickt diese zur **ALU**, die damit arbeitet. Zur Ausführung dieser Operation werden **Steuersignale** ausgesendet. Die Operationen werden koordiniert, indem eine Folge von Impulsen verwendet wird, die von einem Quarzkristall, dem **Taktgeber**, erzeugt werden. Die Geschwindigkeit eines Computers wird gemessen nach der Anzahl der Impulse, die der Taktgeber in der Sekunde erzeugt. Ein **Megahertz** (MHz) entspricht einer Million Impulse pro Sekunde.

Unterbrecher: Mehrmals in der Sekunde sendet die Steuereinheit ein Signal aus, das man **Unterbrechung** nennt. Es bewirkt, daß die Zentraleinheit ihre Arbeit unterbricht. Der Unterbrecher veranlaßt den Computer, sein Unterbrechungsprogramm auszuführen. Damit wird z. B. geprüft, ob inzwischen irgendwelche Tasten auf der **Tastatur** angeschlagen wurden.



Das Betriebssystem

Das **Betriebssystem** enthält alle Anweisungen, die die Zentraleinheit benötigt, um ihre verschiedenen Aufgaben zu erfüllen. Dieses Betriebssystem besteht aus speziellen Programmen, die oft im **Festwertspeicher** (**ROM**) gespeichert sind. Jeder Computerhersteller hat sein eigenes Betriebssystem, das die Fähigkeiten des Computers bestimmt. Manche Betriebssysteme, wie z. B. das **CP/M** (**C**ontroll **P**rogram/**M**ikroprozessor), lassen sich in Computern verschiedener Herstellerfirmen einsetzen; dadurch kann man auf diesen Computern auch die gleichen Programme ablaufen lassen.

Urlader: Befindet sich das Betriebssystem des Computers noch nicht im ROM-Speicher, dann muß es geladen werden, sonst kann der Computer nicht einmal die einfachsten Aufgaben ausführen. Das Laden erfolgt durch den Urlader, ein kurzes Programm, das schon im ROM gespeichert ist.

Mikroprozessor-Typen

Es gibt eine beschränkte Anzahl verschiedener Typen von Mikroprozessoren (siehe Seite 108). Alle diese Typen werden in den gebräuchlichsten Mikrocomputern verwendet. Vom Typ des Mikroprozessors hängt es ab, in welchem Umfang die Zentraleinheit den **Befehlsvorrat** (siehe Seite 134) an **Maschinencode** verarbeiten kann und mit welcher Geschwindigkeit sie arbeitet. Es gibt inzwischen verbesserte Versionen dieser Mikroprozessoren, die schneller arbeiten und mehr Befehle verstehen können.

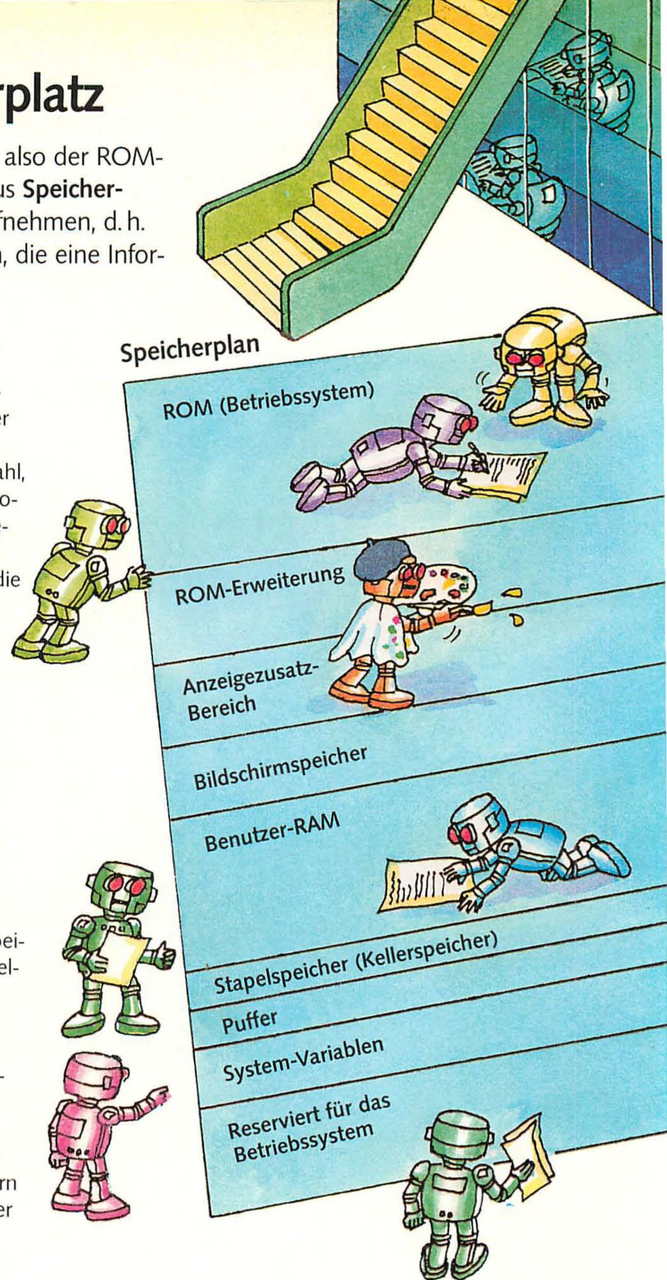
6502 und **6510**: Diese Typen (Hersteller: Mostek) gehören zu den am meisten verwendeten Mikroprozessoren. Der 6502 wird im PET, im VC-20 und im Apple II verwendet. Der 6510 ist eine verbesserte Version des 6502 und wird im Commodore 64 verwendet.

Z 80 und **Z 80 A** (Hersteller: Zilog): Der Z 80 A ist eine verbesserte Version des Z 80 und wird z. B. in Sinclair-/Timex-Computern verwendet.

6800 und **6809** (Hersteller: Motorola): Der 6809 ist eine verbesserte Version des 6800 und wird im Dragon und im Tandy/Radio Shack-Farbcomputer verwendet.

Speicher und Speicherplatz

Der gesamte Speicherplatz des Computers, also der ROM-Speicher und der RAM-Speicher, besteht aus **Speicherzellen**. Jede Speicherzelle kann ein **Byte** aufnehmen, d. h. eine Gruppe von acht elektrischen Signalen, die eine Information darstellen.



Speicheradresse: Jede Speicherzelle hat eine Nummer, die sogenannte **Adresse**, so daß der Computer einmal gespeicherte Informationen wiederfinden kann. Jede Adresse besteht aus einer 16-Bit-Binärzahl, wenn 16 Adreßleitungen vorhanden sind. Ein Mikrocomputer kann dann bis zu 65 536 (64 K) verschiedene Speicherzellen haben, die von 0 bis 65 535 numeriert sind – dies ist die höchste Dezimalzahl, die mit 16 Binärzeichen dargestellt werden kann.

Seite: Dieser Begriff wird für eine aufeinanderfolgende Gruppe von Speicherzellen verwendet. Die Größe einer Seite ist von Computer zu Computer verschieden. Bei den meisten Mikrocomputern besteht sie aus 256 Speicherzellen, bei anderen Computern kann diese Zahl auch 512 oder 1 024 (1 KB) betragen.

Der Speicherplan

Die verschiedenen Bereiche des Speichers sind für unterschiedliche Zwecke eingeteilt. Unter einem Speicherplan versteht man einen Plan, in dem die einzelnen Bereiche und ihre Adressen angegeben sind. Jedes Computer-Handbuch enthält normalerweise einen solchen Speicherplan. Eine Auswahl der verschiedenen Bereiche ist abgebildet und wird im folgenden näher erklärt.

Benutzer-RAM: Dies ist der Bereich des RAM-Speichers, in dem man Programme und Daten speichern kann. Man kann nicht den gesamten RAM-Speicher benutzen, weil der Computer einen Teil des Speichers für sich selbst benötigt, sobald man ihn einschaltet.

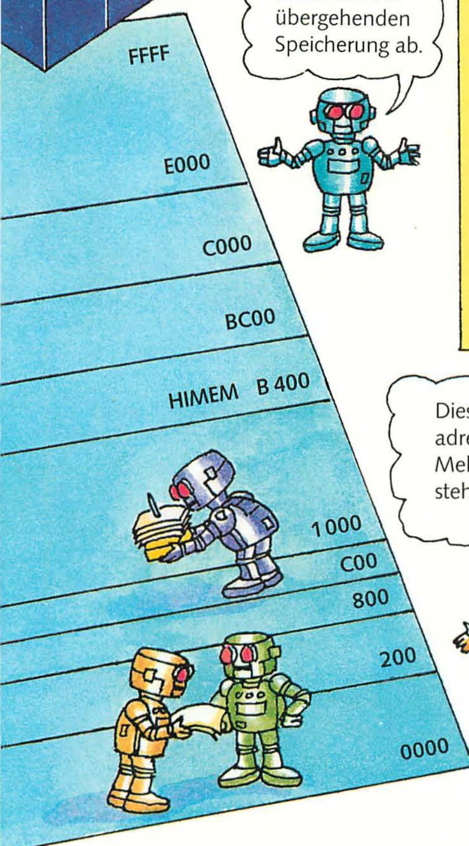
Bildschirmspeicher: Dies ist eine Gruppe aufeinanderfolgender Speicherzellen, die man auch **Grafikspeicher** nennt. Sie dienen zur Speicherung der Information, die auf dem Bildschirm ausgegeben werden soll. Jede Speicherzelle wird an einer bestimmten Stelle auf dem Bildschirm dargestellt. Die Anzeige nennt man auch Speicherorientiert. Man kann die Anzeige abändern, indem man die Inhalte der einzelnen Zellen ändert.

Anzeigezusatz-Bereich: Diese Speicherzellen enthalten besondere Informationen, die man **Anzeigezusätze** nennt, so z. B. Informationen über **Grafik** oder **Zeichen**, die auf dem Bildschirm gezeigt werden sollen, d. h. über ihre Farbe oder die Art ihrer Schrift (z. B. negativ).

Puffer: Dieser Bereich des Speichers wird als Zwischenspeicher benutzt, um Informationen zu speichern, die entweder zwischen verschiedenen Teilen des Computers oder zwischen dem Computer und einem Peripheriegerät transportiert werden. Beispiele für Puffer sind Tastaturpuffer und Kassettenpuffer.

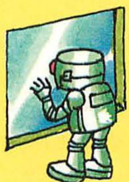
Reserviert für das Betriebssystem: Dieser Speicher wird vom **Betriebssystem** (siehe Seite 113) des Computers benutzt, um die Informationen zu speichern, die der Computer für seine Arbeit benötigt. Bei manchen Computern sind **Puffer**, **Stapelspeicher** und **System-Variablen** in diesem Bereich des Speicherplans vereinigt.



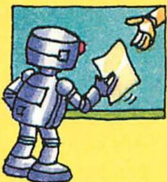


PEEK und POKE

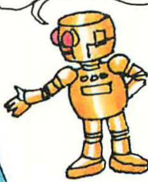
Wenn man direkt mit bestimmten Speicherplätzen arbeiten will, kann man die **BASIC**-Befehle **PEEK** und **POKE** benutzen, zusammen mit der **Adresse** des Speichers, den man ansprechen möchte.



PEEK: PEEK bedeutet „hineinschauen", d. h. der Computer soll den Inhalt einer **Speicherzelle** lesen. Man kann jede Speicherzelle lesen lassen.



POKE: Bewirkt, daß der Computer Informationen in einer Speicherzelle speichert. Dies kann man in jeder **RAM**-Speicherzelle tun, man sollte jedoch vermeiden, dies bei **Systemvariablen** zu tun, da man damit das System durcheinanderbringen könnte. Sollte dies einmal vorkommen, kann man den Computer wieder zu seiner normalen Arbeitsweise zurückführen, indem man ihn aus- und wieder einschaltet.



HIMEM: Darunter versteht man die Grenze eines Speichers, auch **RAMTOP** genannt. Dies ist die höchste Adresse im Benutzer-RAM, d. h. die höchste Adresse, in der man eingegebene Informationen speichern kann.

Warteschlange: Dieser Ausdruck wird für solche Bereiche des Speichers benutzt, in denen die Daten nach dem Prinzip gespeichert werden: Was zuerst hereinkommt, geht zuerst hinaus.

Systemvariablen: Diesen Bereich des Speichers benutzt der Computer, um sich Informationen zu „merken", die sich ständig ändern, wie z. B. die Position des Cursors auf dem Bildschirm oder die Adresse des nächsten freien Raums im **Benutzer-RAM.**

Stapelspeicher: In diesem Bereich des Speichers werden Daten in ganz besonderer Weise gespeichert. Er arbeitet nach dem Prinzip: Was zuletzt hereinkommt, geht zuerst hinaus, d. h. die zuletzt eingegebene Information wird zuerst wieder ausgegeben. Stapelspeicher können ganz verschiedenen Zwecken dienen: Der **GOSUB-Stapelspeicher** speichert z. B. die Nummer der Programmzeile, zu der der Computer nach Ausführung eines **Unterprogramms** innerhalb des Programms zurückspringt, der **Rechner-Stapelspeicher** speichert in einer längeren Rechnung die Zwischenergebnisse, und den **Maschinen-Stapelspeicher** oder auch **Prozessor-Stapelspeicher** benutzt die **Zentraleinheit**, um verschiedene Informationen zu speichern, z. B. die Stelle in einem Programm, an die nach einer **Unterbrechung** (siehe Seite 113) zurückzukehren ist.

Weitere Speicherarten

Virtueller Speicher: Manche Computer können Informationen auf **Diskette** speichern und wieder auffinden, so als ob die Diskette Teil des Speichers wäre, d. h. ohne dazu Befehle vom Benutzer zu benötigen. Dieser Sonderspeicher heißt virtueller Speicher.

RAM-Diskette: Es handelt sich dabei gar nicht um eine Diskette, sondern um eine Anzahl von **RAM-Chips**, die der Computer jedoch wie eine Diskette behandelt. Er benötigt einen Befehl, um Informationen abzurufen; die Bearbeitung geschieht allerdings schneller als bei einer Diskette.

Magnetblasenspeicher: Darunter versteht man besondere Chips, die Informationen als winzige Magnetblasen speichern. Man kann damit sehr viel mehr Daten speichern als mit einem Silizium-Chip, auch dann, wenn der Strom abgeschaltet ist. Der Magnetblasenspeicher arbeitet jedoch langsamer als ein normaler Speicher.

Peripheriegeräte

Peripheriegeräte sind Geräte, die man an einen Computer anschließen kann. Man nennt sie gelegentlich auch Zusatzgeräte. Die beiden häufigsten Peripheriegeräte sind der **Kassettenrecorder** und das **Diskettenlaufwerk**, die beide dazu dienen, Daten und Programme zu speichern.
Um ein Peripheriegerät an einen Computer anzuschließen, braucht man eine **Schnittstelle**. Dieses Bauteil steuert die Signale zwischen dem Computer und dem Peripheriegerät und wandelt sie so um, daß beide sich untereinander „verstehen".
Ein **Eingabe-/Ausgabegerät** ist ein Peripheriegerät, das man benutzt, um Daten in einen Computer einzugeben, oder das Daten benutzt, die von einem Computer ausgegeben werden, um bestimmte Aufgaben durchzuführen. Im folgenden werden einige der gebräuchlichsten vorgestellt.

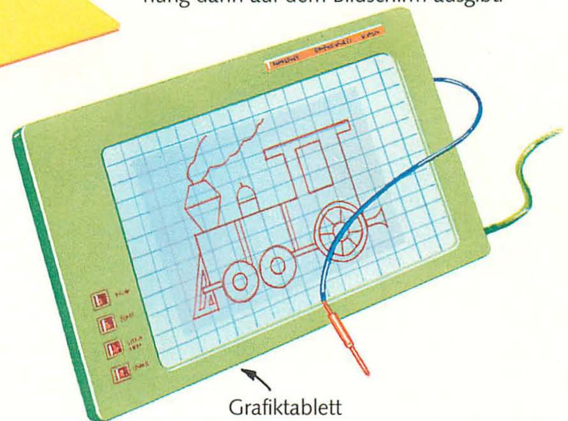
Digitalisierer: Er besteht aus einem drehbaren Arm, mit dem man über Zeichnungen und Grafiken fährt. Das Gerät schickt dann Informationen zum Computer, damit diese als Zeichnungen auf dem Bildschirm ausgegeben werden.



Lichtgriffel: Mit diesem kugelschreiberähnlichen Instrument, das man an den Computer anschließen kann, kann man direkt auf den Bildschirm zeichnen oder durch Berühren des Bildschirms bestimmte Gegenstände auswählen.



Grafiktablett: Wird manchmal auch **Digitalisierer** genannt. Man kann auf dieser Tafel zeichnen, dabei werden Signale an den Computer gesandt, der diese Zeichnung dann auf dem Bildschirm ausgibt.

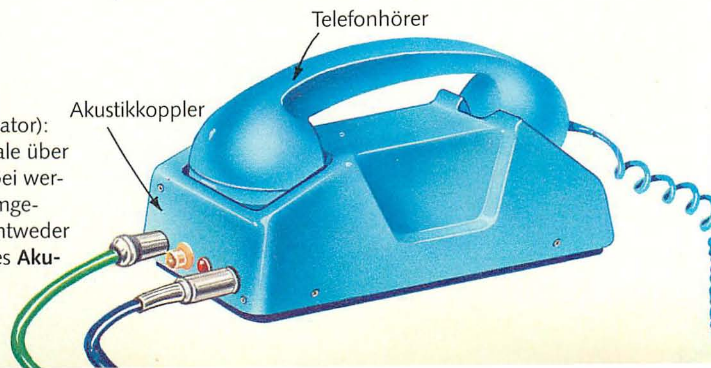


Maus: Dies ist ein kleines Gerät, oft mit einer Kugel, das man zur Angabe benutzt, in welche Richtung der Computer einen Zeiger auf dem Bildschirm bewegen soll.

Analog-Digital-Wandler: Die Daten, die von Peripheriegeräten zum Computer gesendet werden, werden in Form unterschiedlich starker elektrischer Spannungen transportiert. Ein Analog-Digital-Wandler „übersetzt" diese Spannungsschwankungen in Impuls- oder Kein-Impuls-Signale, die man als **digitale Signale** bezeichnet; nur diese Signale kann der Computer verarbeiten. Ein ähnliches Gerät, nämlich der **Digital-Analog-Wandler**, erledigt diese Aufgabe in umgekehrter Richtung.

Akustikkoppler: Darunter versteht man ein tragbares **Modem**, das an jeden Computer angeschlossen werden kann und das man zusammen mit dem Telefon benutzt. Man legt den Telefonhörer in den Koppler, der dann die Signale zum Computer und vom Computer umwandelt.

Modem (Abkürzung für **Mo**dulator/**Dem**odulator): Mit einem Modem kann man Computersignale über das Telefonnetz senden und empfangen. Dabei werden die Computersignale in Telefonsignale umgewandelt und umgekehrt. Ein Modem kann entweder fest eingebaut sein, oder es hat die Form eines **Akustikkopplers** (siehe rechts).



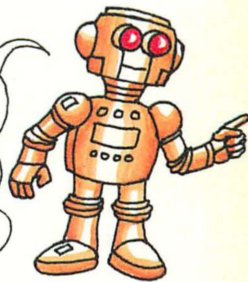
Ausdruck
Drucker
Feuerknopf
Drehregler
Steuerknüppel (Joystick)
Plotter
Farbige Zeichenstifte

Drucker: Gerät, das die vom Computer kommenden Daten ausdruckt. Es gibt verschiedene Arten von Druckern (siehe Seite 120). Einige können auch **Grafiken** anfertigen. Die gedruckte Datenausgabe nennt man **Ausdruck**.

Steuerknüppel (Joystick): Den Steuerknüppel benutzt man, um Gegenstände auf dem Bildschirm zu bewegen. Dieses Gerät wird hauptsächlich bei Computerspielen verwendet. Die meisten dieser Steuerknüppel haben einen „**Feuerknopf**", mit dem Gegenstände auf dem Computerbildschirm „abgeschossen" werden können.

Drehregler: Ein knopfähnliches Gerät, das genau wie ein **Steuerknüppel** eingesetzt wird, um Gegenstände auf dem Bildschirm zu bewegen (insbesondere bei Ballspielen).

Ein Peripheriegerät ist **on-line**, wenn es an einen Computer angeschlossen ist und Signale zum Computer senden und vom Computer empfangen kann. Es ist **off-line**, wenn es ausgeschaltet bzw. nicht mit dem Computer verbunden ist.

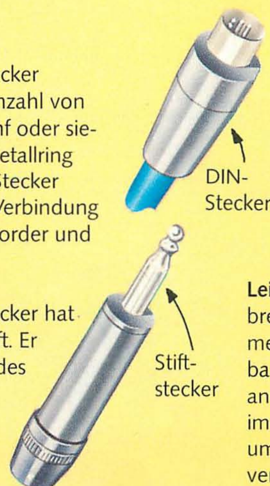


Plotter: Dieses Zeichengerät empfängt Signale vom Computer und zeichnet Grafiken, Diagramme oder andere Abbildungen auf Papier.

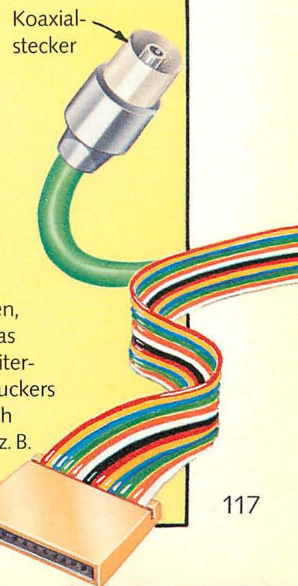
Stecker und Kabel

DIN-Stecker: Diese Stecker haben eine gewisse Anzahl von Stiften, gewöhnlich fünf oder sieben, die von einem Metallring umgeben sind. Diese Stecker werden meistens zur Verbindung zwischen Kassettenrecorder und Computer eingesetzt.

Stiftstecker: Dieser Stecker hat nur einen dickeren Stift. Er dient zur Verbindung des Computers mit einer Stromquelle.



Koaxialstecker: Dient als Verbindung eines Computers zum Fernsehgerät oder zum **Monitor** (siehe Seite 104). Das Kabel hat eine Leitung in der Mitte, die von einer Isolierschicht umgeben ist; darüber liegt eine weitere Leitung. Die mittlere Leitung führt in den Stift in der Mitte des Steckers.



Leiterband: Es besteht aus einem flachen, breiten und flexiblen Kunststoffkabel, das mehrere parallele Leitungen enthält. Leiterband dient z. B. zum Anschluß eines Druckers an einen Computer, es wird jedoch auch im Inneren des Computers verwendet, z. B. um die Tastatur mit der Platine zu verbinden.

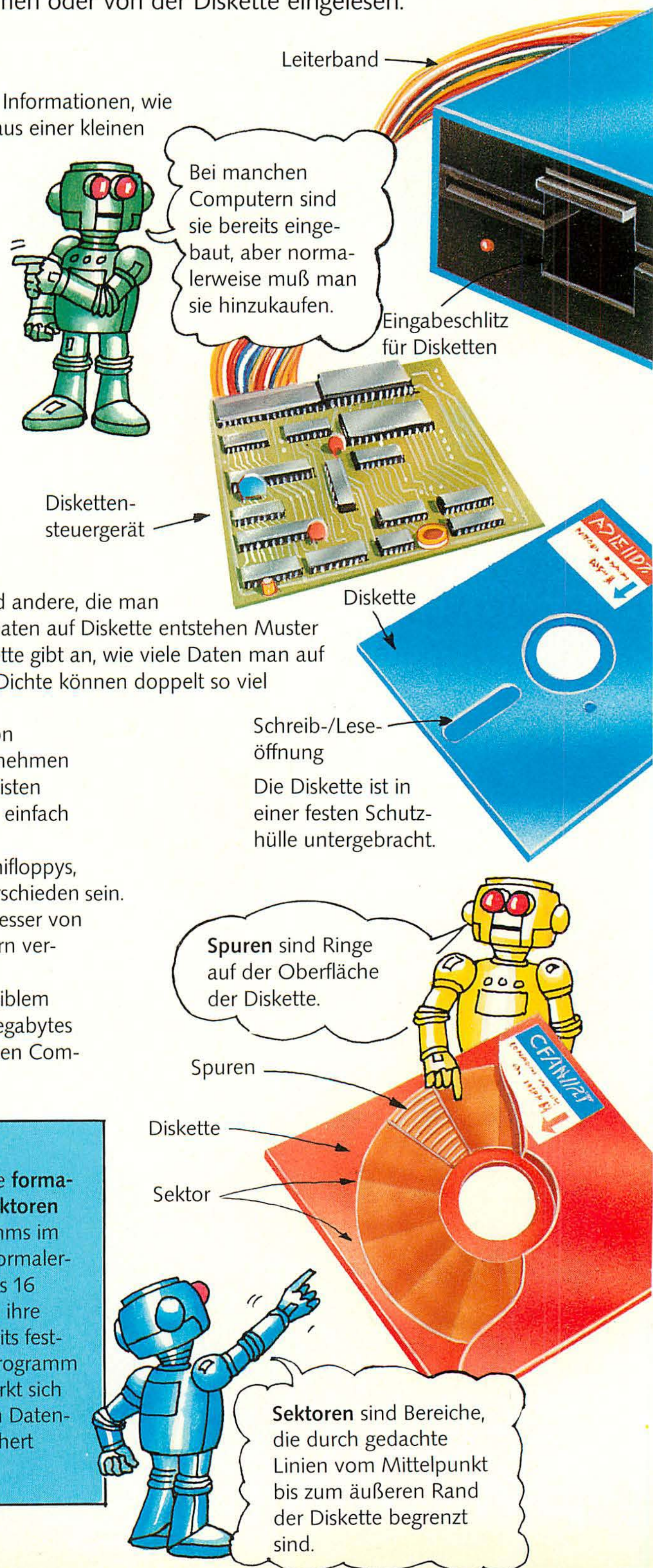
Disketten und Laufwerke

Disketten, auch **Floppy disks** genannt, sind dünne Kunststoffscheiben, auf denen der Computer Programme und Daten speichern kann. Im **Diskettenlaufwerk** werden die Programme und Daten auf die Diskette aufgenommen oder von der Diskette eingelesen.

Diskettensteuergerät: Dieses Gerät gibt dem Computer Informationen, wie das Diskettenlaufwerk arbeiten soll; es besteht zumeist aus einer kleinen **Platine** mit **Chips**.

Disketten-Schnittstelle: Dieses Gerät steuert die Signale zwischen Computer und Diskettenlaufwerk.

Disketten-Betriebssystem (abgekürzt **DOS**): Darunter versteht man eine Zusammenstellung von Programmen, die dem Computer sagt, wie er Informationen von einer Diskette abrufen bzw. auf einer Diskette speichern soll. Bei manchen Computern befindet sich das Disketten-Betriebssystem bereits im **ROM**, andere Computer verfügen über ein kurzes Programm, das ausschließlich dazu dient, das DOS von einer Diskette zu laden.



Verschiedene Arten von Disketten

Es gibt Disketten, die nur einseitig zu bespielen sind, und andere, die man doppelseitig bespielen kann. Bei der Speicherung von Daten auf Diskette entstehen Muster aus magnetischen Punkten. Die **Schreibdichte** der Diskette gibt an, wie viele Daten man auf einer Diskette speichern kann. Disketten mit doppelter Dichte können doppelt so viel speichern wie Disketten mit einfacher Dichte.

Minifloppys: Darunter versteht man kleine Disketten von 5,25 Zoll Durchmesser, die aber wenigstens 100 KB aufnehmen können, je nach der **Schreibdichte** der Diskette. Die meisten Computer verwenden Minifloppy-Disketten, die jedoch einfach als Disketten oder Floppy disks bezeichnet werden.

Mikrofloppys: Diese Disketten sind noch kleiner als Minifloppys, die Menge der unterzubringenden Daten kann sehr verschieden sein.

Standardfloppys: Diese Disketten haben einen Durchmesser von 8 Zoll und werden auf den meisten größeren Computern verwendet.

Festplatte: Diese Platten bestehen aus festem, nicht flexiblem Material. Eine typische Festplatte kann wenigstens 5 Megabytes speichern. Diese Platten werden hauptsächlich auf großen Computern verwendet.

Formatieren einer Diskette

Bevor man eine Diskette in Gebrauch nimmt, muß sie **formatiert** werden, d. h. sie muß in **Spuren** und einzelne **Sektoren** aufgeteilt werden. Dies erfolgt mit Hilfe eines Programms im **Disketten-Betriebssystem**. Eine **Minifloppy** enthält normalerweise 40 Spuren bzw. 80 bei Doppelspur sowie 10 bis 16 Sektoren. Manche Disketten sind **hardsektoriert**, d. h. ihre Sektoren sind durch entsprechende Indexlöcher bereits festgelegt. **Softsektorierte** Disketten werden durch das Programm in Sektoren und Spuren unterteilt. Der Computer merkt sich die Spuren und Sektoren, damit er die verschiedenen Datenblöcke oder Programme, die auf der Diskette gespeichert sind, schneller finden kann.



Schreib-/Lesekopf: Er berührt die Oberfläche der sich drehenden Diskette. Dabei werden Informationen entweder abgerufen oder eingegeben.

Schreib-/Leseöffnung: So nennt man das Loch in der Hülle der Diskette, durch das der Schreib-/Lesekopf entweder Informationen eingibt oder entnimmt. Die Diskette selbst darf nie aus der Schutzhülle entfernt werden!

Mikrolaufwerk: Ein kleines Diskettenlaufwerk, mit dem man Mikrofloppy-Disketten abspielen kann.

Doppellaufwerk: Ein Diskettenlaufwerk, das zwei **Schreib-/Leseköpfe** besitzt und zwei Disketten aufnehmen kann.

Schreibschutz-Aussparung: Dies ist eine kleine Kerbe in der Plattenhülle, die anzeigt, daß man die Diskette bespielen kann. Ist sie z. B. mit Klebeband abgedeckt oder überhaupt nicht vorhanden, dann bedeutet dies bei Minifloppys, daß man die Platte nicht bespielen kann. Diese Maßnahme verhindert, daß man vorhandene Daten aus Versehen überspielt.

Laufwerkangabe: Die meisten Mikrocomputer können mehr als ein Diskettenlaufwerk in Betrieb nehmen. Damit der Computer weiß, welches Laufwerk er in Betrieb nehmen soll, benötigt er den Befehl der Laufwerkangabe.

Dateien

Unter einer Datei versteht man ein Programm- oder Datenpaket, das unter einem bestimmten Namen auf einer Diskette oder einer Kassette gespeichert ist.

Datendatei: Eine Datei, die nur Daten enthält und keine Programme. Bestehen die Daten aus Text, d. h. Wörtern und Sätzen, dann spricht man von einer **Textdatei**.

Datei mit wahlfreiem (oder: **direktem**) **Zugriff**: Diese Datei ist so beschaffen, daß der Computer jede beliebige Information an einer beliebigen Stelle der Datei sofort findet. Das Gegenstück dazu ist eine **serielle Datei**, bei der der Computer die gesamte Datei von Anfang an durchsuchen muß.

Datensätze und **Datenfelder**: Um Daten leichter zu finden und auszuwechseln, kann man eine Datei in Datensätze und Datenfelder aufteilen. So würde man z. B. eine Datei, in der Einzelheiten über verschiedene Computertypen gespeichert sind, in Datensätze für jeden Typ aufteilen. Diese Datensätze könnte man nochmals in Datenfelder unterteilen, wobei sich dann in jedem Datenfeld eine bestimmte Information (z. B. die Speichergröße) über einen bestimmten Computertyp befinden würde.

Diskettendatei-Verzeichnis: Dies ist die Aufstellung des Computers über die Stellen auf der Diskette, an denen sich eine Datei oder ein Programm befindet. Dieses Datenverzeichnis beansprucht normalerweise einige **Spuren** am Anfang der Diskette.

Reservediskette: Eine Diskette, die Kopien der Daten enthält, die schon auf einer anderen Diskette gespeichert sind. Das ist eine Sicherheitsmaßnahme für den Fall, daß die Originaldiskette aus Versehen gelöscht oder beschädigt wird.

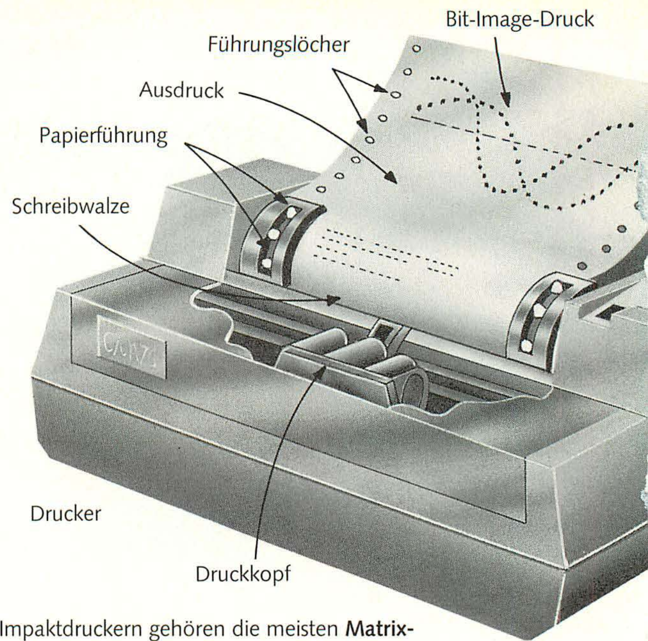
Geschützte Datei: In diesem Fall lädt der Computer die erwünschten Daten nur dann, wenn man das richtige Codewort eingibt. Damit wird sichergestellt, daß nur berechtigte Personen Zugang zu diesen Daten haben.

Dateiverwaltung: Darunter versteht man den Umgang des Computers mit Dateien, also das Suchen, Lesen, Schreiben und Löschen von Daten.

Diskettendatei-System: Darunter versteht man lediglich eine Sammlung von Dateien, die auf einer oder mehreren Disketten untergebracht ist – praktisch ein „computerisierter" Karteikasten.

Drucker

Ein **Drucker** ist ein Gerät, das Programme und Daten, die der Speicher des Computers ausgibt, auf Papier ausdruckt. Es gibt viele verschiedene Sorten von Druckern, die unterschiedliche Druckmethoden verwenden. Es gibt **serielle Drucker** oder **Zeichendrucker**, die ein **Zeichen** nach dem anderen drucken. Ein **Zeilendrucker** druckt auch jedes Zeichen extra, jedoch so schnell, daß man meinen könnte, er drucke eine ganze Zeile auf einmal.

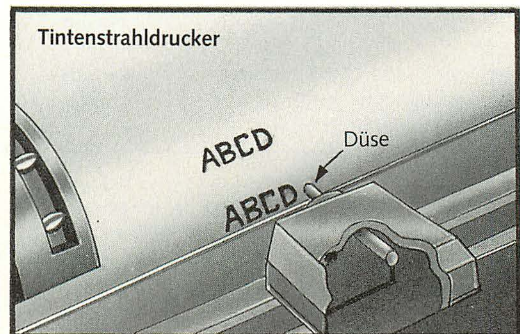


Verschiedene Arten von Druckern

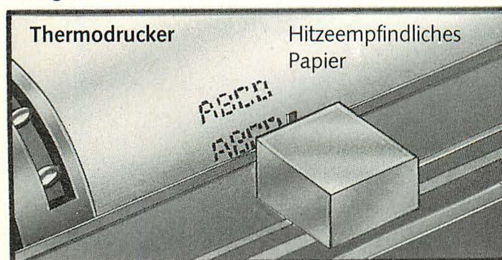
Impaktdrucker: Dies sind alle Drucker, die Zeichen drucken, indem sie ein Farbband gegen das Papier drücken, so wie eine übliche Schreibmaschine. Zu den Impaktdruckern gehören die meisten **Matrix-Drucker**, die **Typenraddrucker** und **Typenkorbdrucker**. Sie unterscheiden sich durch die Form ihres Druckkopfes.



Matrix-Drucker: Der Druckkopf besteht aus einem Satz feiner Nadeln (daher auch **Nadeldrucker**). Ein Zeichen wird gedruckt, indem die Nadeln in einer ganz bestimmten Anordnung gegen das Farbband schlagen.

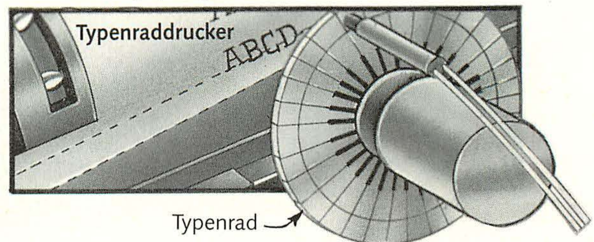


Tintenstrahldrucker: Der Druck der Zeichen erfolgt, indem sehr feine, genau berechnete Tintentröpfchen auf das Druckpapier gebracht werden.



Thermodrucker: Diese Art von **Matrix-Drucker** benötigt ein spezielles hitzeempfindliches Papier, in das der Druckkopf die Zeichen „einbrennt".

Funkendrucker (oder **Elektroerosionsdrucker**): Funktioniert ähnlich wie ein **Thermodrucker**. Das Papier ist jedoch aluminiumbeschichtet, und die Zeichen werden von elektrischen Funken eingebrannt.



Typenraddrucker: Hier erfolgt der Druck durch ein Typenrad, bei dem jedes Zeichen am Ende eines kleinen „Arms" angebracht ist. Um ein bestimmtes Zeichen zu drucken, dreht sich die Scheibe so lange, bis das gewünschte Zeichen erreicht ist; dann wird es gegen das Farbband geschlagen.

Typenkorbdrucker: Funktioniert ähnlich wie ein **Typenraddrucker**, die Zeichen sind jedoch am Rand eines nach oben gebogenen Typenarms angebracht.

Ausdruck: Darunter versteht man die Ausgabe eines Computers, die er auf Papier ausdruckt. Für Ausdrucke werden verschiedene Papierarten verwendet, manche Drucker können jedoch mehrere Papierarten verarbeiten (siehe nächste Seite).

Bit-Image-Betrieb: Manche Drucker können sowohl Text als auch **Grafiken** ausdrucken. Bei Bit-Image-Betrieb druckt der Drucker Punkte, die zu einer Grafik zusammengesetzt werden.

Schreibwalze: Die Rolle oder der Zylinder, um den das Papier geführt wird.

Zeilenschaltung: Darunter versteht man den Vorgang, bei dem der Druckkopf zum Druck der nächsten Zeile an den Ausgangspunkt zurückgeführt wird.

Druckkopf: Dieser Teil des Druckers sorgt für den Druck. Das kann z. B. ein Typenrad sein oder eine Düse, die Tintentröpfchen abgibt.

Führungsstifte: Das sind die Stifte, die auf einem Rad sitzen und in die Führungslöcher des Papiers (siehe nächste Seite) greifen. Sie transportieren so das Papier.

Bidirektionales Drucken: Bei dieser Druckart bewegt sich der Druckkopf auf dem Papier hin und her, indem er die erste Zeile von links nach rechts druckt, die nächste Zeile von rechts nach links usw. Dadurch wird viel Zeit gespart. Bidirektionale Drucker können sogar entscheiden, welches die schnellste Möglichkeit ist, die nächste Zeile zu drucken, je nach der Länge der Zeile, die gerade beendet wurde, bzw. der Länge der folgenden Zeile.

Druckerpuffer: Ein spezieller Teil des **Speichers** im Computer oder im Drucker, der die auszudruckenden Daten „vorrätig hält". Dies ist notwendig, da der Computer die Daten sehr viel schneller ausgibt, als der Drucker sie drucken kann.

Zeichen pro Sekunde: Das Maß für die Geschwindigkeit eines Druckers.

Die Zeichen

Schriftart: Darunter versteht man das Aussehen der Zeichen eines Druckers.

PRINT

PRINT

Breite Zeichen: Diese Zeichen sind doppelt so breit wie die üblichen Zeichen.

PRINT

Schmale Zeichen: Diese Zeichen sind schmaler als die üblichen Zeichen.

$Y_2=X_1+X_3$

$5^2+3^3=52$

Tiefgestellte und **hochgestellte Zeichen**: Darunter versteht man kleine Buchstaben oder Ziffern, die tiefer oder höher stehen als die übrigen Zeichen.

Proportionaldruck: Manche Drucker können den Raum zwischen den einzelnen Zeichen verändern, so daß z. B. ein „I" weniger Platz braucht als ein „N". Bei Druckern, die das nicht können, nehmen alle Zeichen den gleichen Raum ein.

PRINT

PRINT

Fettdruck: Diese Zeichen sind dunkler gedruckt als üblich, so daß sie hervorstechen.

Richtige und **falsche Unterlängen**: Unterlängen sind die unteren „Enden" eines Buchstabens, wie z. B. beim „p" und beim „g". Manche Drucker drucken richtige Unterlängen aus, d. h. die Unterlängen reichen unter die Druckzeile. Andere Drucker drucken die Buchstaben so, daß die Unterlängen auf der Druckzeile enden.

graphic

graphic

Druckerpapier

Es gibt im wesentlichen zwei Arten von Druckerpapier: einzelne **lose Bogen** oder das sogenannte **Endlospapier**. Beim Endlospapier unterscheidet man zwei Arten: **gefaltetes Papier** und **Rollenpapier**.

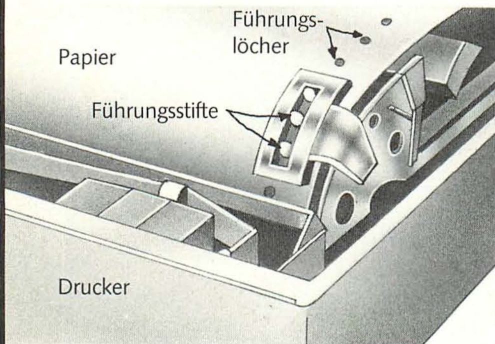
Gefaltetes Papier: Dieses Papier ist durch Perforationslinien in einzelne, aber zusammenhängende Bogen aufgeteilt. Die Bogen werden zuerst in der einen und dann in der anderen Richtung gefaltet und als Block gestapelt.

Rollenpapier: Dieses Papier wird in Rollenform geliefert und benötigt einen speziellen Papierhalter.

Papiertransport

Der Vorgang, bei dem das Papier während des Drucks vorwärtsbewegt wird, heißt Papiertransport.

Reibungsvorschub: Auf diese Weise wird das Papier zwischen der Schreibwalze und einer Anzahl kleinerer Rollen festgehalten.



Stiftvorschub (auch **Traktor** oder **Führungsstiftvorschub**): Am Ende der Schreibwalze befindet sich ein Rad mit hervorstehenden **Führungsstiften**. Diese Stifte greifen genau in die entsprechenden Löcher des Papiers und transportieren es so durch den Drucker.

Zeilenvorschub: Damit wird das Papier am Ende einer Druckzeile weiter nach oben gerückt.

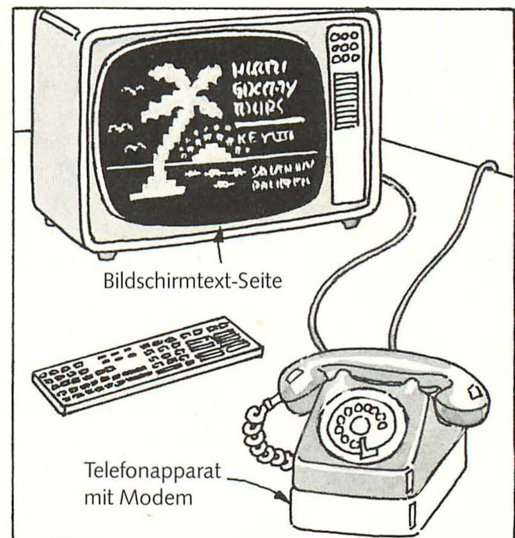
Rück-Vorschub: Manche Drucker können das Papier sowohl nach oben wie nach unten bewegen. Das ist besonders beim Druck von **Grafiken** von Vorteil.

Datenbanken

Unter einer **Datenbank** versteht man einen großen Informationsspeicher, der in einem Großcomputer gespeichert ist. Die Informationen sind entweder nur einer ausgewählten Personengruppe oder auch öffentlich zugänglich. Datenbanken mit beschränktem Zugang enthalten z. B. Informationen über Steuerbescheide, Straftäter oder geheime Regierungsdaten. Datenbanken, die jedermann zugänglich sind, enthalten die verschiedensten Informationen, angefangen von Sportergebnissen über Kochrezepte und Wetterberichte bis zu Börsenkursen und Wirschaftsnachrichten.

Wie man eine Datenbank benutzt

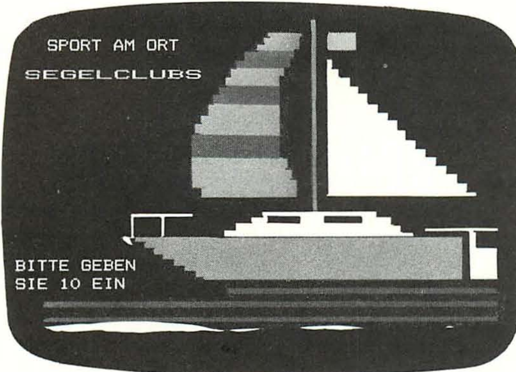
Es gibt verschiedene Möglichkeiten, Informationen von einer Datenbank abzurufen. Das kann mit Hilfe einer Kabelverbindung zwischen dem eigenen Computer und dem der Datenbank geschehen, die Informationen können jedoch auch über das Telefonnetz gesendet werden. Kommen sie über die Telefonleitung und werden sie auf einem Fernsehbildschirm dargestellt, so spricht man von **Bildschirmtext**.



Bildschirmtext: Darunter versteht man Informationen, die als Signale über das Telefonnetz gesendet werden und als Text auf einem Fernsehbildschirm erscheinen. Um Bildschirmtext zu empfangen, braucht man zwei Zusatzgeräte, das **Modem** (siehe Seite 116), das die Signale entschlüsselt, und einen **Decoder**, der sie entschlüsselt. Das Modem kann entweder an ein Fernsehgerät angeschlossen sein, das den Decoder enthält, es kann aber auch mit einem Computer verbunden sein.

Netzwerke

Datenfernübertragung: Darunter versteht man die Übermittlung von Informationen zwischen zwei Computern über große Entfernungen mit Hilfe von Telefonkabeln. Auf diese Weise gesendete Computerprogramme heißen **Tele-Software**.

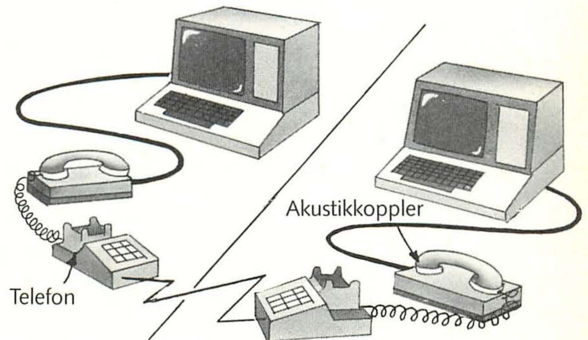


Hier ein Beispiel für Informationen, die als **Bildschirmtext** gesendet werden.

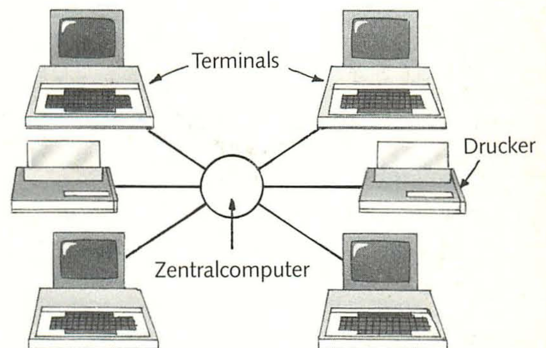


Videotext: Darunter versteht man Informationen, die von einer zentralen Stelle parallel zu den Signalen der normalen Fernsehprogramme gesendet werden. Zum Empfang braucht man keinen Computer, das Fernsehgerät muß jedoch einen eingebauten **Decoder** haben, der die ankommenden Signale in Text umwandelt, die man als **Seiten** bezeichnet. Jede Seite hat ihre eigene Nummer, die man mit Hilfe der Fernbedienung anwählen kann.

Unter einem **Netzwerk** versteht man eine Gruppe von Computern, die über Telefonkabel oder durch eigene Kabel miteinander verbunden sind, so daß sie untereinander Informationen austauschen oder **Peripheriegeräte** gemeinsam benutzen können.



Bei einem **Heimcomputer-Netzwerk** werden Nachrichten und Programme über das Telefonnetz in einzelne Haushalte gesendet. In den USA und in Großbritannien gibt es bereits solche Heimcomputernetze, bei denen es sogar möglich ist, nur bestimmten Teilnehmergruppen Zugang zu diesem Netz zu verschaffen.



Zur Verbindung von Wirtschaftsunternehmen ist ein **Netzwerk** sehr nützlich. Ein solches Netzwerk kann verschiedene Computer miteinander verbinden oder auch einen zentralen Computer und eine bestimmte Anzahl von Terminals und andere Peripheriegeräte. Ein **Terminal** kann z. B. nur aus **Tastatur** und Bildschirm bestehen und dazu benutzt werden, Informationen mit dem Zentralcomputer auszutauschen, oder es kann seine eigenen Daten bearbeiten; in diesem Fall spricht man von einem **intelligenten Terminal**. Bei einem System mit einem zentralen Computer und einer gewissen Anzahl von Terminals spricht man von einem **Mehrplatzsystem**, da mehrere Personen den Computer zur gleichen Zeit benutzen können. Man spricht dann auch von **Teilnehmerbetrieb**.

Grafik

Der Begriff **Grafik** wird für alles verwendet, was der Computer an Bildern und Diagrammen ausgibt. Die Grafik kann auf dem Bildschirm erscheinen oder auf einem **Drucker** oder einem **Plotter** ausgegeben werden.

Wie der Computer Grafiken herstellt

Grafiken, wie Buchstaben und Ziffern, erscheinen auf dem Bildschirm durch Aufhellung winziger Punkte, sogenannter **Bildpunkte**. Die Bildqualität eines Computers hängt stark davon ab, wie viele dieser Bildpunkte er auf dem Bildschirm aufhellen kann.

Bildpunkte: Darunter versteht man die Punkte auf dem Bildschirm, die aufgehellt werden, um ein „Bild" entstehen zu lassen. Manche Computer können die Größe dieser Punkte verändern und Bilder unterschiedlicher Schärfe erzeugen. Man spricht dabei von **Auflösung** (siehe unten).

Hochauflösende Grafik: Ein Computer, der eine große Anzahl kleiner Bildpunkte aufhellen kann, ist in der Lage, sehr detaillierte Bilder herzustellen; man nennt das hochauflösende Grafik. Ein üblicher Heimcomputer hat ungefähr 40 000 Bildpunkte, es gibt jedoch auch Computer, die ein besonderes, hochauflösendes Grafikprogramm haben und daher sehr viel mehr Bildpunkte aufhellen können.

Geringauflösende Grafik: Darunter versteht man Bilder, die aus einer verhältnismäßig kleinen Anzahl sehr großer Bildpunkte hergestellt werden. Die Formen dieser Bilder sind etwas unförmig und unscharf.

Grafik-Betriebsart: Bei manchen Computern muß man zur Grafik-Betriebsart übergehen, bevor man Bilder erzeugen kann. Die meisten Computer haben verschiedene Grafik-Betriebsarten mit unterschiedlichen Auflösungsgraden oder verschiedenen Farbmöglichkeiten.

Fenster: Manche Computer kann man so programmieren, daß sie an einer beliebigen Stelle des Bildschirms eine freie Fläche erzeugen. In diese Fenster kann man innerhalb einer Grafik Text schreiben.

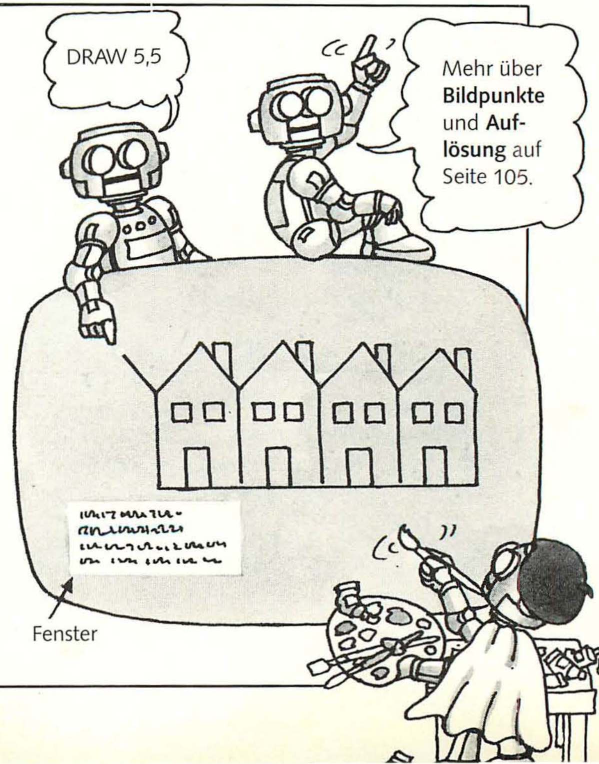
Anzeigetypen: Die Typen einer Grafikanzeige sind z. B. Farben oder Negativschrift oder Blinkzeichen. Diese Informationen sind im Speicher als Anzeige-Bytes gespeichert.

Farbpalette: Darunter versteht man die Anzahl der Farben, die ein Computer auf den Bildschirm bringen kann.

Grafikbefehle

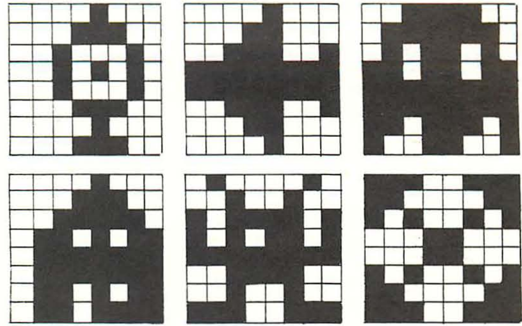
Mit bestimmten Befehlen, z. B. DRAW und PLOT, kann man den Computer veranlassen, eine Grafik anzufertigen. Diese Befehle sind jedoch von Gerät zu Gerät unterschiedlich.

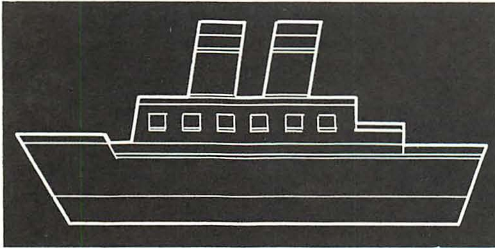
PLOT: Dies bedeutet, daß ein Bildpunkt aufgehellt werden soll. Um das zu erreichen, muß man dem Computer einen Grafikbefehl und die **Koordinaten** des **Bildpunktes** eingeben. Die Koordinaten sind zwei Zahlen, die die Position des Punktes auf dem Bildschirm angeben. Die X-Koordinate gibt die Anzahl der Bildpunkte in der Breite des Bildschirms an, die Y-Koordinate die Anzahl der Punkte in der Höhe. **Absolute Koordinaten** werden von der oberen oder unteren linken Bildschirmecke aus gemessen, relative Koordinaten vom letzten Punkt aus, der auf dem Bildschirm gezeichnet wurde.



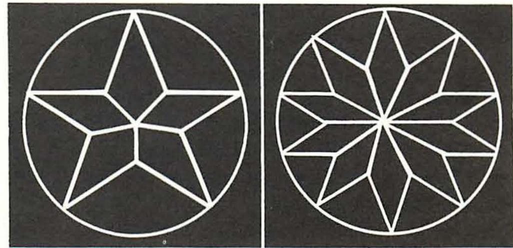
Verschiedene Arten von Grafik

Grafikzeichen: Darunter versteht man kleine Flächen, die denselben Raum wie ein normales Zeichen einnehmen, d. h. wie ein Buchstabe oder eine Ziffer. Bei manchen Computern befinden sie sich auf den Tasten, und man kann sie direkt benutzen, um Bilder herzustellen. Die Menge der Grafikzeichen, die auf einem bestimmten Computer verfügbar ist, nennt man **Grafikzeichen-Satz**. Man kann auch seine eigenen Grafikzeichen herstellen, dann spricht man von Benutzer-definierten Grafikzeichen.





Liniengrafik: Bei dieser Methode werden Bilder dadurch erzeugt, daß man den Computer so programmiert, daß er Linien von bestimmten Punkten aus zu anderen Punkten auf den Bildschirm zeichnet. Der Computer hellt dann automatisch alle Punkte dieser Linie auf.



Schildkrötengrafik: Dies ist eine Methode zur Erzeugung von Bildern, die durch die Bewegung einer Pfeilspitze auf dem Bildschirm entstehen. Die Pfeilspitze heißt **Schildkröte**. Man kann sie in bestimmten Winkeln drehen und bestimmte Entfernungen wandern lassen, wobei sie entweder eine Linie zeichnet oder nichts.

Animationsgrafik: Dies sind Bilder oder Formen, die über den Bildschirm wandern. Der Bewegungseffekt wird dadurch erreicht, daß man die Formen an einer bestimmten Stelle zeichnet, sie dann löscht und an einer anderen Stelle des Bildschirms nochmals zeichnet.
Es gibt besondere Grafikprogramme, die alle Positionen ausarbeiten, die zwischen zwei Punkten liegen, und sie automatisch anzeigen. Diesen Vorgang nennt man **Einbetten**.

Sprites: Dies sind Figuren in einer Grafikanzeige, die man so programmieren kann, daß sie als eine Einheit an einem beliebigen Punkt des Bildschirms erscheinen und sich umherbewegen. Sie sind oft größer als Grafikzeichen und können sich so bewegen, daß sie vor oder hinter anderen Sprites erscheinen.

Computer Aided Design (Zeichnen mit Computerhilfe), abgekürzt **CAD**: CAD-Programme dienen dazu, Konstrukteuren und Architekten bei der Anfertigung schwieriger technischer Zeichnungen zu helfen. Die Programme dafür werden in zunehmendem Maße auch zur Produktgestaltung in der Industrie verwendet.

Dreidimensionale Grafik: Darunter versteht man Abbildungen auf dem Bildschirm, die dreidimensional zu sein scheinen. Die einfachsten Bilder dieser Art sehen so aus, daß alle Linien des dargestellten Gegenstands ausgezeichnet sind, auch dann, wenn sie bei einem festen Körper eigentlich unsichtbar sein müßten.



Programmiersprachen

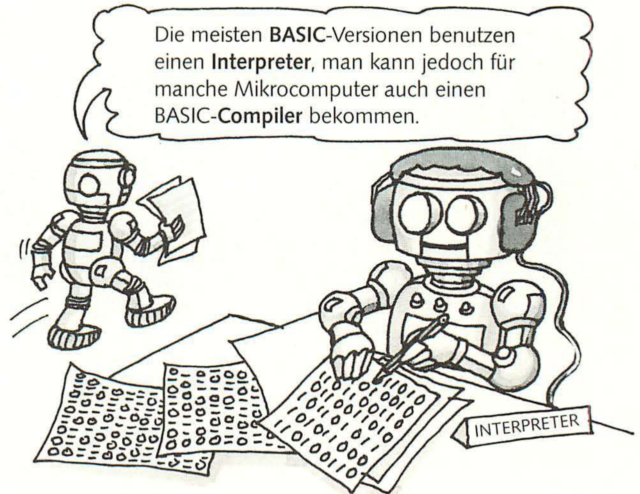
Eine **Programmiersprache** besteht aus Wörtern, Ziffern und Symbolen, die man verwenden muß, um einem Computer Befehle zu erteilen. Es gibt viele verschiedene Programmiersprachen, von denen manche für ganz spezielle Aufgaben entwickelt wurden. Man spricht von **höheren Programmiersprachen**, wenn darin Wörter verwendet werden, die unserer eigenen Sprache ähnlich sind. **Einfache Programmiersprachen** benutzen den **Binärcode** des Computers (siehe Seite 110). Es gibt zwei Haupttypen von einfachen Programmiersprachen: den **Maschinencode** und die **Assemblersprache**; in beiden Fällen spricht man jedoch von Programmieren im Maschinencode (siehe Seite 134).

Interpreter und Compiler

Die höheren Programmiersprachen müssen erst in den Maschinencode des Computers übersetzt werden. Dazu dient ein spezielles Programm, das entweder ein **Interpreter** (auch: Interpreter) oder ein **Compiler** ist. Der Unterschied zwischen beiden besteht in der Art, wie sie die Übersetzung des Programms vornehmen.

Läßt man ein Programm ablaufen, dann übersetzt der Interpreter einen Befehl nach dem anderen und führt ihn genauso aus. Ein Compiler übersetzt nur alle Befehle und führt sie nicht aus. Compilierte Programme laufen sehr viel schneller ab als interpretierte.

Im **ROM**-Speicher der meisten Heimcomputer befindet sich ein **BASIC**-Interpreter. Will man eine andere Programmiersprache benutzen, dann kann man normalerweise einen Interpreter oder Compiler für diese Sprache in den Speicher des Computers laden.



Quellcode: Dies ist die Sprache, in der ein Programm geschrieben ist.
Objektcode: In diesen Code wird das Computerprogramm übersetzt.

Höhere Programmiersprachen

BASIC (Abkürzung für **B**eginner's **A**ll-purpose **S**ymbolic **I**nstruction **C**ode): Dieses ist die gängigste Sprache dieser Art, obwohl BASIC-Programme nicht so übersichtlich sind und nicht so schnell ablaufen wie Programme, die in anderen Sprachen geschrieben sind. Die verschiedenen Computerhersteller verwenden verschiedene Versionen von BASIC, die sich in einigen Wörtern und Regeln etwas voneinander unterscheiden. Diese Versionen heißen **Dialekte**. Als Beispiel seien genannt: Atari-BASIC, Microsoft-BASIC und Sinclair-BASIC.

```
10 REM - EIN BASIC-PROGRAMM
20 PRINT "WIE HEISSEN SIE ?";
30 INPUT N$
40 PRINT "GUTEN TAG, ";N$;
50 PRINT "NETT, SIE ZU SEHEN"
```

Es gibt das sogenannte **„Mini"-BASIC**, eine Version von BASIC, in der man nicht alle Befehle dieser Sprache verwenden kann. Demgegenüber gibt es jedoch auch ein **erweitertes BASIC**, das mehr Befehle enthält und eine vielseitigere Verwendung gestattet.

```
function ZAEHLER (var PAR, GRENZE : integer)
                                  : integer;
begin
        if PAR > GRENZE then
                ZAEHLER := GRENZE
        else
                if PAR < 1 then
                        ZAEHLER := 1
                else
                        ZAEHLER := PAR;
end;
```

Pascal: Diese Programmiersprache ist nach dem französischen Mathematiker und Naturwissenschaftler Blaise Pascal benannt, der im 17. Jahrhundert lebte. Die Sprache existiert seit den sechziger Jahren. Man kann damit eine Vielzahl von Problemen behandeln, und die Programme sind **strukturiert**, d. h. sie sind logisch in Blöcken angeordnet. Diese Sprache zu erlernen, ist jedoch recht schwierig.

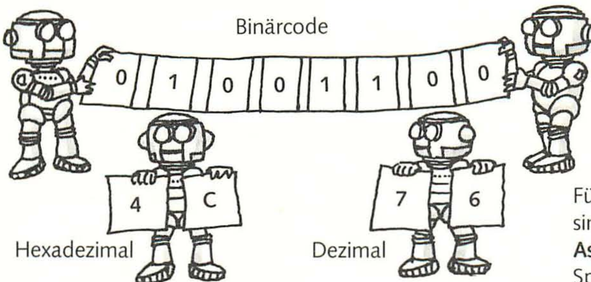
PILOT: Diese Sprache wurde entwickelt, um Programme für computerunterstütztes Lehren und Lernen zu schreiben. Diese Programme dienen der Wissensvermittlung, z. B. im Schulunterricht.

Einfache Programmiersprachen

Jede Aufgabe, die ein Computer ausführt, besteht aus vielen kleinen Einzelschritten, die durch eine oder mehrere Gruppen von acht Signalen, die **Bytes**, dargestellt werden. In einer **höheren Programmiersprache** wird durch einen Programmierbefehl eine Kette von Operationen für eine bestimmte Aufgabe ausgelöst, in einfacheren Programmiersprachen muß man dem Computer jedoch für jede Operation einen Befehl erteilen. Programme in einfacher Programmiersprache laufen bis zu zehnmal oder noch schneller ab als Programme in einer höheren Programmiersprache.

Maschinencode

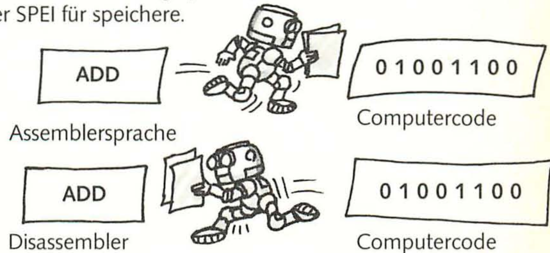
Der Begriff Maschinencode oder Maschinensprache wird manchmal dazu benutzt, um den **Binärcode** des Computers (siehe Seite 110) zu bezeichnen, in den letztlich alle Programme umgewandelt werden müssen. Ganz allgemein bezieht sich dieser Begriff jedoch auf Programme, die in der Hex-Form dieses Binärcodes geschrieben sind. **Hex** ist eine Kurzform



für Hexadezimal. Darunter versteht man ein Zahlensystem, das 16 Zeichen benutzt – die Ziffern 0 bis 9 und die Buchstaben A bis F. Dieses System funktioniert wie das Dezimalsystem, man zählt jedoch bis fünfzehn, ehe der Sprung zur nächsten Stelle erfolgt.

Assemblersprache

Dies ist eine weitere Möglichkeit, ein Programm in einer einfachen Programmiersprache zu schreiben. Die Befehle, die dem Computer für jede Operation gegeben werden, sind in einem **mnemonischen Code** geschrieben. Darunter versteht man eine bestimmte Abkürzungssprache, z. B. ADD für addiere oder SPEI für speichere.



Für Programme, die in Assemblersprache geschrieben sind, braucht man ein spezielles Programm, das man **Assembler** nennt. Dieses Programm wandelt die Sprache in den Computercode um. Der Assembler kann bereits in den Computer eingebaut sein, manchmal muß man ihn jedoch separat eingeben. Ein **Disassembler** verwandelt das Programm im Speicher des Computers in die Assemblersprache zurück, falls man es z. B. auf dem Bildschirm ausgeben möchte.

```
: CUBE           ( VOM ERSTEN WERT IM STACK)
  DUP            ( ARGUMENT ZUM DRUCKEN
                   BEREITSTELLEN)
  CR 3 .R        ( DRUCKEN)
  DUP DUP * *    ( KUBIEREN)
  6 .R           ( DAS ERGEBNIS DRUCKEN)
;
```

FORTH: Diese Sprache wurde für den Gebrauch auf kleinen Computern entwickelt und ermöglicht es, eigene Operationen, Programmierwörter oder Strukturen zu bilden. FORTH-Programme belegen sehr wenig Speicherplatz.

```
1-02-00-UPDATE-FILES SECTION.
1-02-00-ENTER
      READ VALID TRANSACTIONS
         AT END MOVE 1 TO EOT-FLAG.
      IF TRANSACTIONS-PROCESSED
         NEXT SENTENCE
      ELSE
         IF HEADER
                   NEXT SENTENCE
         ELSE
                   PERFORM 2-02-00-
                   TRANSACTIONS-PROCESS.
1-02-99-EXIT.
         EXIT.
```

COBOL (Abkürzung für **CO**mmon **B**usiness **O**riented **L**anguage): Dies ist eine Sprache, die für wirtschaftliche Zwecke entwickelt wurde und sich sehr gut für die Verwaltung von Dateien u. ä. eignet.

```
      REAL FUNCTION VERDOPPELT (ZAHL)
C
C     DIESE FUNKTION VERDOPPELT DAS ARGUMENT
C
      REAL ZAHL
      VERDOPPELT = 2.0 * ZAHL
      RETURN
      END
```

FORTRAN (Abkürzung für **FOR**mular **TRAN**slation): Diese Sprache wurde in den fünfziger Jahren erfunden und ist die am häufigsten verwendete Sprache für wissenschaftliche und mathematische Aufgaben. Die bekannteste Version heißt **FORTRAN 77**.

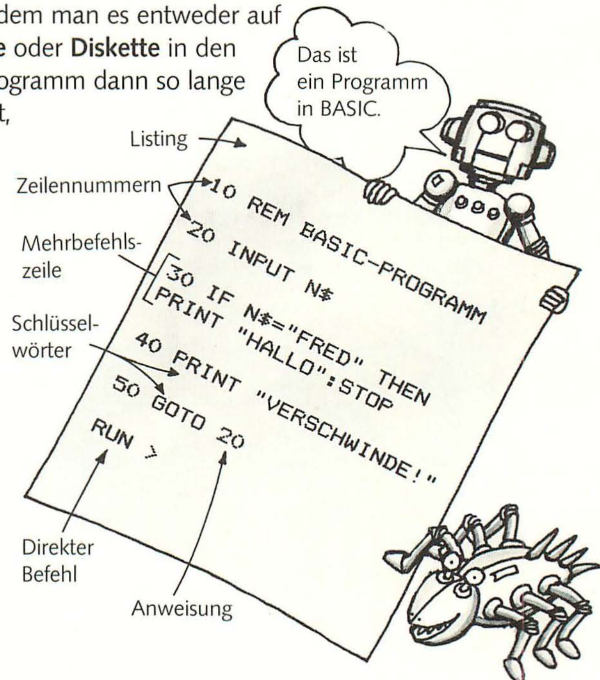
```
TO RAD
REPEAT 8 (RIGHT 45 RECHTECK)
RIGHT 180
FORWARD 50
END
```

LOGO: Diese Sprache wurde entwickelt, um bei Kindern im Vor- und Grundschulalter mathematisches Verständnis zu schaffen. In dieser Sprache werden Befehle benutzt, mit denen man einen kleinen Roboter namens „Schildkröte" bewegen kann.

Das Programm

Ein **Programm** ist eine Liste von Befehlen. Sie ist in einer Sprache geschrieben, die der Computer „versteht" und die ihm sagt, wie er eine bestimmte Aufgabe auszuführen hat. Außer den eingegebenen Befehlen benötigt der Computer noch Informationen, mit denen er arbeiten kann. Diese Informationen heißen **Daten**.

Man gibt dem Computer ein Programm ein, indem man es entweder auf der **Tastatur** eintippt oder es von einer **Kassette** oder **Diskette** in den Computer lädt. Der Computer speichert das Programm dann so lange in seinem **Speicher**, bis man ihn dazu auffordert, das Programm ablaufen zu lassen, d. h. die Befehle auszuführen.



Listing: Die vollständige Wiedergabe des Programms, entweder als Anzeige auf dem Bildschirm oder als **Ausdruck** auf Papier.

Dokumentation: Darunter versteht man alle Anmerkungen, die erklären, wie ein Programm arbeitet.

Zeilennummer: Die Nummer am Anfang einer jeden Zeile eines **BASIC**-Programms.

Anweisung: Ein anderes Wort für Befehl.

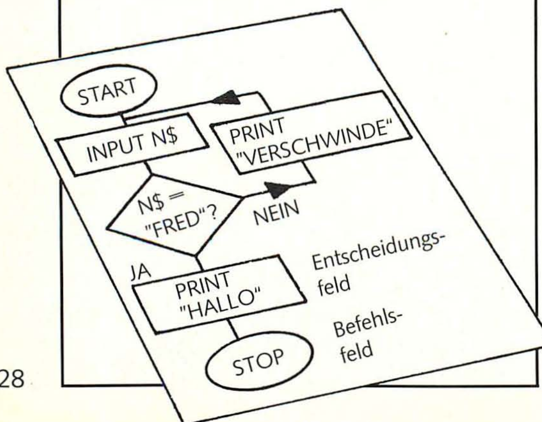
Mehrbefehlszeile: Eine Programmzeile, in der mehr als eine **Anweisung** enthalten ist.

Schlüsselwörter: Die Wörter, aus denen eine Programmiersprache besteht. In BASIC sind dies z. B. die Wörter PRINT, INPUT, IF... THEN. Jedes Wort löst eine Kette von Operationen im Inneren des Computers aus.

Direkter Befehl: Darunter versteht man eine Anweisung, die nicht Teil des Programms ist. Sie wird sofort ausgeführt.

Programmplanung

Flußdiagramm: Dieses Diagramm dient dazu, das Programm für einen Computer zu planen. Es zeigt die Schrittfolgen, die zur Lösung eines Problems erforderlich sind.



Fehler

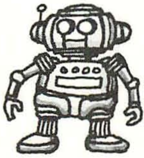
Einen Fehler im Programm nennt man **Bug** (engl. für Wanze). Liegt der Fehler in der Benutzung der **Programmiersprache**, dann spricht man von einem **Syntax-Fehler**. Dies kann z. B. durch Eintippen falscher Buchstaben entstehen. Sobald ein Fehler in einem Programm auftaucht, gibt der Computer eine **Fehlermeldung** aus. Sie enthält normalerweise eine Information darüber, um was für einen Fehler es sich handelt. In manchen Fällen verweist der Computer jedoch lediglich auf das Computer-Handbuch.
Die Beseitigung von Fehlern nennt man **Debugging**.

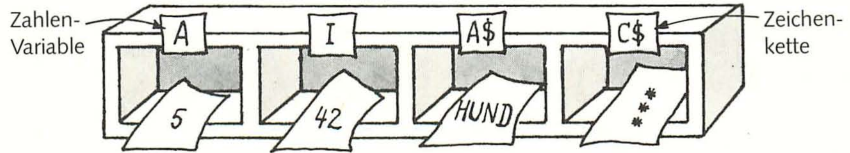
Absturz: Unterbricht der Computer plötzlich seine Arbeit und reagiert nicht auf Tastatureingaben, dann sagt man, er ist abgestürzt. Ein solcher Absturz kann verschiedene Ursachen haben. Den Computer wieder in Gang zu setzen, gelingt normalerweise nur dadurch, daß man ihn aus- und wieder einschaltet oder den **RESET**-Knopf drückt.

Spannungsabfall: Durch unregelmäßige Stromzufuhr können Fehler im Computerprogramm entstehen, in manchen Fällen kann sogar das Programm vollständig gelöscht werden.

Variablen

Eine Variable ist ein „Name", mit dem man eine Information in einem Computerprogramm bezeichnet. Beispiel: Die Anweisung LET C = 8 gibt der Variablen C den Wert 8. Der Wert der Variablen kann sich innerhalb eines Programms mehrmals ändern.





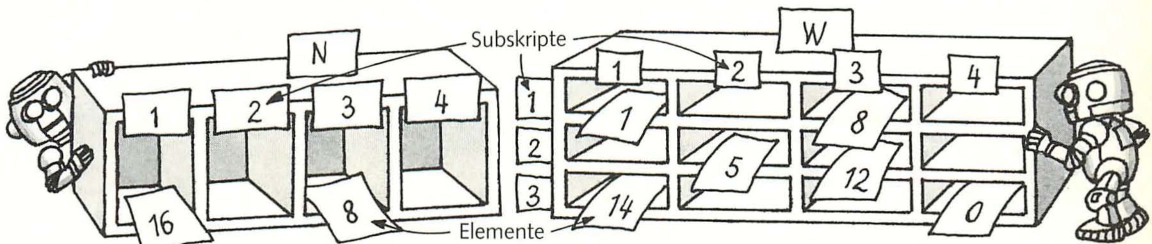
Zahlen-Variable: Darunter versteht man eine Variable, deren Wert aus Zahlen besteht. In **BASIC** bezeichnet man Variablen gewöhnlich mit einem Buchstaben, z. B. A oder I.

Ganzzahlige Variablen: Die wechselnden Werte einer ganzzahligen Variablen sind stets ganze Zahlen. In BASIC wird diese Art Variable mit dem %-Zeichen versehen.

Fließkomma-Variablen: Der Wert, den man diesen Variablen gibt, besteht aus Fließkommazahlen oder **reellen Zahlen** (also Zahlen mit Stellen hinter dem Komma).

Zeichenketten oder **String-Variablen**: Diese Variablen können für Buchstaben und Zeichen ebenso verwendet werden wie für Zahlen. In BASIC werden sie mit $ gekennzeichnet, z. B. C$. Dies spricht man aus: „C-String" oder „C-Dollar". Ein **Null-String** enthält nichts.

Feld: Unter einem Feld versteht man eine Gruppe von zusammenhängenden **Daten**, die unter einer bestimmten Bezeichnung im Speicher abgelegt sind. Die Bezeichnung und die Positionsnummer im Feld charakterisieren jedes Element dieser Daten. Die Positionsnummer nennt man **Subskript**. Ein Zeichenfeld enthält Buchstaben oder Zeichen; in BASIC wird es mit $ bezeichnet.



Eindimensionales Feld: Ein solches Feld nennt man auch **Vektor** und versteht darunter ein Feld, in dem jede Information nur ein **Subskript** besitzt. So würde man z. B. vier Zahlen, die in einem Feld N gespeichert sind, folgendermaßen bezeichnen: N(1), N(2), N(3), N(4). Dies könnten z. B. die Altersangaben von vier Menschen sein. Ihre Namen würde man in einer Zeichenkette speichern, z. B. M$.

Zweidimensionales Feld: In einem solchen Feld, das man auch **Matrix** nennt, werden die Daten in Reihen und Spalten angeordnet, und jede Information hat zwei Subskripte: die Reihen- und die Spaltennummer. So bedeutet z. B. W(2,3) das dritte Element in der zweiten Reihe eines Feldes W.

Wie man mit Variablen umgeht

Bestimmen: Eine Variable bestimmen, bedeutet, ihr einen Wert zuweisen. Eine Anweisung wie z. B. LET A = 40 ist eine **Zuweisungsanordnung**.

Löschen: Dies bedeutet, den Wert der Variablen gleich 0 setzen.

Initialisieren: Dies bedeutet, einer Variablen einen Anfangswert zuweisen.

Ein Feld dimensionieren: Damit werden dem Computer Anzahl und Anordnung der Elemente in einem **Feld** angezeigt. In BASIC erfolgt dies mit dem Befehl DIM sowie mit dem Namen des Feldes und der Anzahl der Elemente, z. B. DIM N(10).

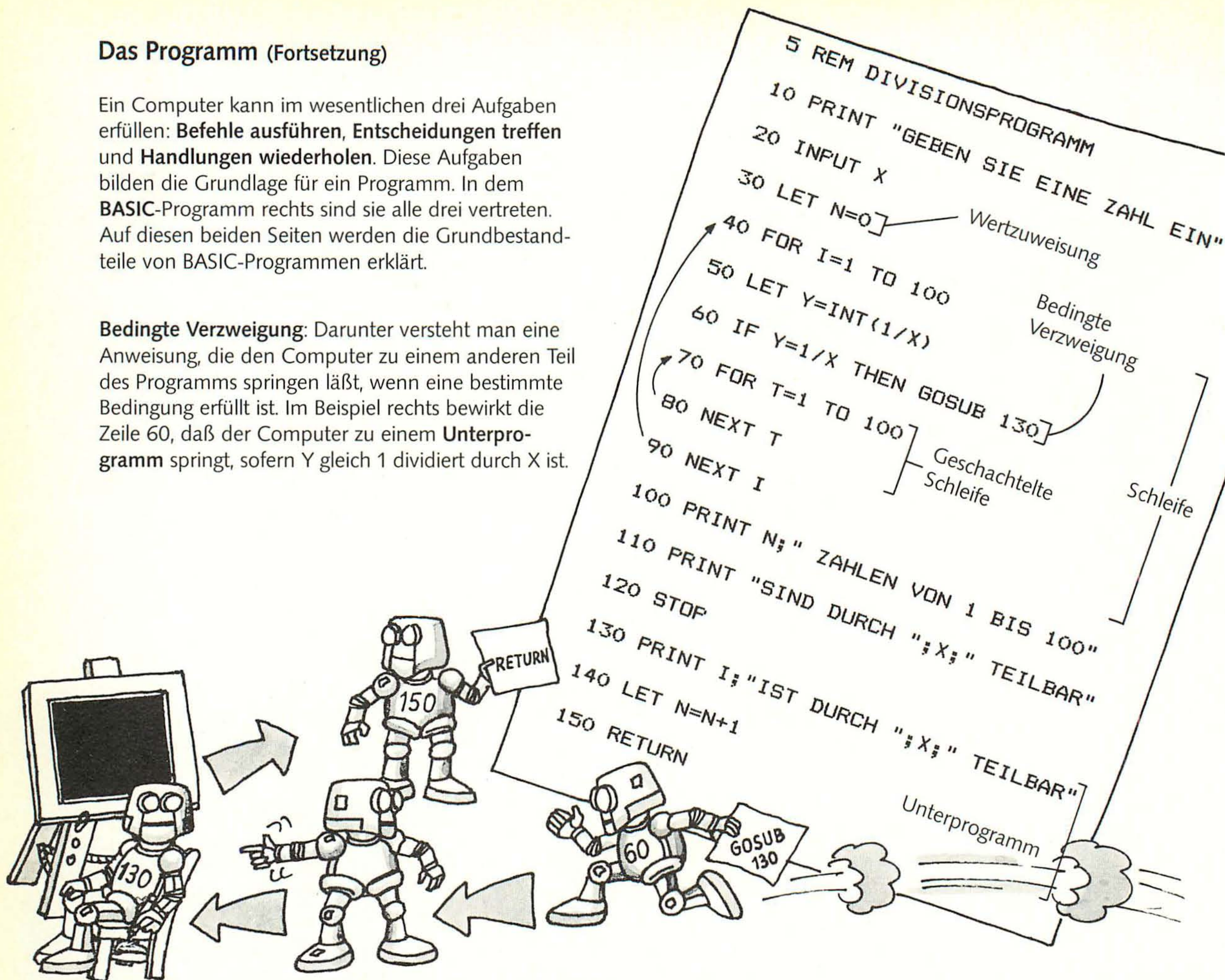
Erhöhen: Man kann den Wert einer Variablen erhöhen, normalerweise um 1, z. B. LET A = A+1.

Vermindern: Man kann der Wert einer Variablen auch vermindern, normalerweise um 1, z. B. LET B = B–1.

Das Programm (Fortsetzung)

Ein Computer kann im wesentlichen drei Aufgaben erfüllen: **Befehle ausführen**, **Entscheidungen treffen** und **Handlungen wiederholen**. Diese Aufgaben bilden die Grundlage für ein Programm. In dem **BASIC**-Programm rechts sind sie alle drei vertreten. Auf diesen beiden Seiten werden die Grundbestandteile von BASIC-Programmen erklärt.

Bedingte Verzweigung: Darunter versteht man eine Anweisung, die den Computer zu einem anderen Teil des Programms springen läßt, wenn eine bestimmte Bedingung erfüllt ist. Im Beispiel rechts bewirkt die Zeile 60, daß der Computer zu einem **Unterprogramm** springt, sofern Y gleich 1 dividiert durch X ist.



Funktionen

Der Computer besitzt eingebaute Anweisungen, die ihm sagen, wie er gewisse Aufgaben auszuführen hat, z. B. mathematische Aufgaben wie addieren, multiplizieren, Quadratwurzeln ziehen, oder andere Aufgaben wie die Bewegung des Cursors an eine bestimmte Stelle auf dem Bildschirm oder das Heraussuchen eines Buchstabens aus einer Zeichenkette. Alle diese Aufgaben nennt man Funktionen. Der Computer führt sie aus, indem man eine bestimmte Taste drückt (z. B. +, *, /) oder indem man ein **Schlüsselwort** in BASIC eingibt, wie z. B. TAB oder MID$.

Bei manchen Computern kann man sogar selbst Funktionen definieren. Mit Hilfe des BASIC-Befehls DEF FN kann man z. B. eine mathematische Funktion definieren. Bei einigen Computern kann man auch bestimmte **Funktionstasten** so programmieren (siehe Seite 103), daß der Computer eine bestimmte Aufgabe immer dann ausführt, wenn diese Taste gedrückt wird.

Zufallszahl: Darunter versteht man eine Zahl, die man nicht vorhersagen kann, z. B. Zahlen, die man beim Würfeln erhält. Der Computer verfügt über ein spezielles Programm, mit dem er Zufallszahlen erzeugen kann.

Routine: Damit bezeichnet man einen in sich abgeschlossenen Teil eines Programms, der eine bestimmte Aufgabe erfüllt.

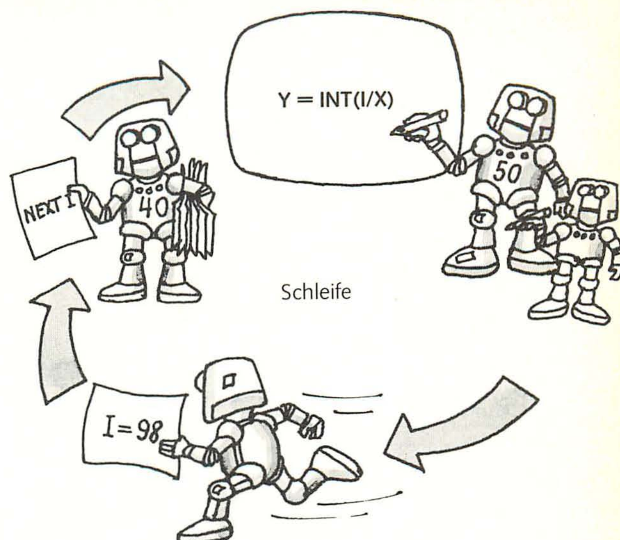
Prozedur: Dies ist ein anderes Wort für **Unterprogramm**, einen bestimmten Abschnitt in einem Programm, der eine spezielle Aufgabe erfüllt.

Druckformatierung: Die spezielle Anordnung von Text auf dem Bildschirm. In BASIC benutzt man z. B. die Schlüsselwörter TAB oder PRINT AT, um dem Computer zu sagen, an welche Stelle das nächste Wort gedruckt werden soll.

Schleife: Sie veranlaßt den Computer, eine Folge von Befehlen so oft, wie gewünscht, zu wiederholen. In **BASIC** erfolgt dies mit den Wörten FOR, TO und NEXT. Die Schleife von Zeile 40 bis 90 im Programm auf der linken Seite veranlaßt den Computer, die Zeilen 50 bis 80 100mal auszuführen.

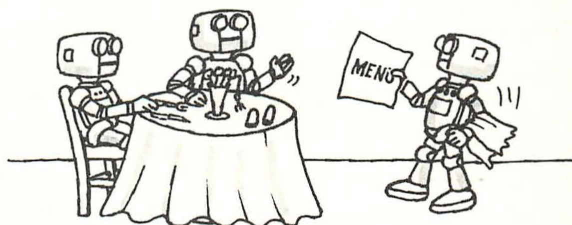
Geschachtelte Schleife: Darunter versteht man eine Schleife innerhalb einer anderen Schleife; sie veranlaßt den Computer, eine Folge von Befehlen jedesmal dann auszuführen, wenn die äußere Schleife wiederholt wird. Im Programm auf der linken Seite ist die geschachtelte Schleife von Zeile 70 bis 80 eine **Verzögerungsschleife**, in der keine weiteren Anweisungen auszuführen sind. Der Computer macht jedoch eine Pause, weil er alle Werte von 1 bis 100 für die Variable T durchzählt.

Unterprogramm: Darunter versteht man einen Programmabschnitt, der eine häufiger vorkommende Aufgabe ausführt. Jedesmal, wenn diese Aufgabe ausgeführt werden soll, wird der Computer durch einen bestimmten Befehl zu dem Unterprogramm geschickt. Am Ende des Unterprogramms kehrt der Computer dorthin zurück, wo er das Programm verlassen hatte. Das Unterprogramm auf der linken Seite beginnt bei Zeile 130. Der RETURN-Befehl in Zeile 150 bewirkt den Rücksprung zu Zeile 70.



Menü: Darunter versteht man eine Auswahlliste, die man in manchen Programmen erhalten kann. Man wählt aus, was man möchte, und der Computer führt den gewünschten Teil des Programms aus. Ein menügesteuertes Programm benutzt im Verlauf seiner Ausführungen viele verschiedene Menüs.





Editieren: Das Abändern und Verbessern eines Programms, indem man einzelne Teile davon nochmals schreibt, dabei Fehler korrigiert und im allgemeinen dafür sorgt, daß das Programm den Wünschen des Benutzers entspricht.

Mischen: Zwei Programme zu einem zusammenlegen.

Operatoren: Darunter versteht man die verschiedenen Zeichen, die ein Computer versteht, z. B. / für dividieren, = für „ist gleich", > für „größer als" und $ für String. (Dieses Zeichen wird zur Kennzeichnung von **Zeichenketten** verwendet.)

Trennzeichen: Dieses Symbol wird zur Trennung von zwei Informationen benutzt. Beispiele für BASIC: Bei der DATA-Anweisung werden Kommas zwischen zwei Elemente gesetzt, bei der PRINT-Anweisung werden die einzelnen Elemente durch Semikolons getrennt.

String-Separierer: Das sind in BASIC die Anführungszeichen, in die eine Zeichenkette eingeschlossen werden muß.

Einen Wert ausgeben: Hat der Computer eine mathematische Berechnung ausgeführt, so gibt er das Ergebnis als Wert aus.

Handhabung von Zeichenketten: Sie dient dazu, Buchstaben aus einer Zeichenkette auszuwählen oder auch zwei Zeichenketten zusammenzulegen, um eine einzige daraus zu machen. In diesem Fall spricht man von **Verbinden**.

Default: Beschreibt das, was der Computer von selbst tut, wenn er keine anderen Anweisungen erhält. So erscheint der Bildschirm in einer bestimmten Ursprungsfarbe, sobald der Computer eingeschaltet wird.

Parameter: Der Wert einer Variablen, der in einem **Unterprogramm** benutzt wird, oder die Beschreibung eines Datenblocks in Zahlenangaben.

Die Software

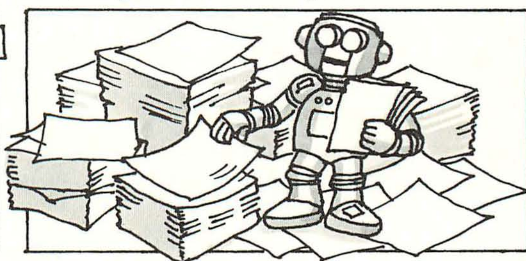
Die **Programme**, mit denen der Computer arbeitet, heißen **Software**. Programme, die bereits in den Speicher des Computers eingebaut sind, heißen **residente Software**. Dazu können das **Betriebssystem** (die Programme, die die Arbeit des Computers steuern) und die **Dienstprogramme** (siehe unten) gehören. Programme, die z. B. von **Kassetten** oder **Disketten** geladen werden oder über die **Tastatur** eingegeben werden, heißen **nicht-residente Software**.

Firmware: Darunter versteht man Programme, die ständig in **ROM-Chips** gespeichert sind, entweder im Speicher des Computers oder in Kassetten, die man in den Computer einstecken kann. Diese Programme heißen Firmware („feste Ware"), weil man sie nicht löschen kann.



Dienstprogramme: Diese Programme werden auch **System-Software** genannt. Es sind Programme, die den Computer in die Lage versetzen, häufig wiederkehrende Aufgaben auszuführen, z. B. **Daten** auf Kassette oder Diskette zu übertragen. Oft sind die wichtigen Dienstprogramme bereits in den Computer eingebaut.

Programmierhilfen: Darunter versteht man eine Sammlung von **Dienstprogrammen**, die auf Kassette oder Diskette gespeichert sind und das Programmieren erleichtern. Eine solche Sammlung kann viele verschiedene Programme enthalten, z. B. ein Programm, das Programmzeilen mit neuen Zahlen versieht oder eines, das ganze Blöcke im Programm löscht.



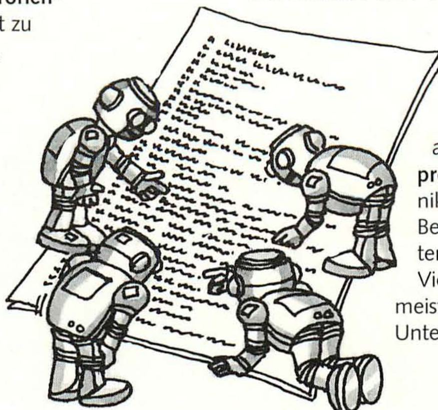
Sortierprogramm: Dieses Programm ordnet Daten, d. h. es erstellt Listen in alphabetischer oder numerischer Reihenfolge. Es gibt viele verschiedene Sortierprogramme, die alle verschiedene Techniken verwenden. Beispiele sind **Bubble sort-**, **Selektions-** und **Aufteilungsprogramme**. Solche Sortierprogramme gehören zu den **Dienstprogrammen** eines Computers.

Programmeigenschaften

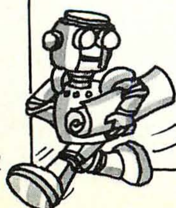
Benutzerfreundlich (auch **benutzerorientiert**): Diese Programme sind leicht zu verstehen und leicht anzuwenden.

Strukturiert: Ein solches Programm ist in abgeschlossene Einheiten unterteilt; jede dieser Einheiten erfüllt eine spezielle Aufgabe. Programme dieser Art sind meist leicht zu lesen und zu verstehen.

Auf den Benutzer abgestimmt: Solche Programme sind geschrieben, um die speziellen Erfordernisse eines Benutzers zu erfüllen.



Interaktiv: Ein interaktives oder **Dialogprogramm** erlaubt Kommunikation zwischen dem Benutzer und dem Computer. Beispiele dafür sind Videospiele wie auch die meisten Programme, die im Unterricht eingesetzt werden.



Übertragbar: Ein Programm ist übertragbar, wenn es ohne Schwierigkeiten auf verschiedenen Computern läuft.

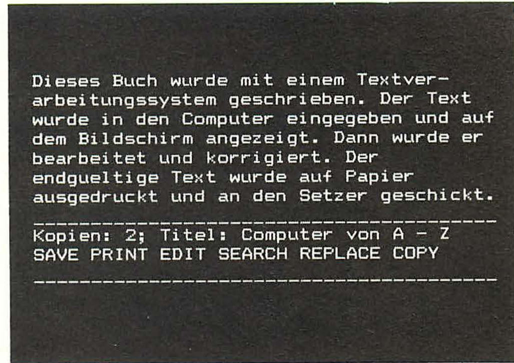
Elegant: So bezeichnet man ein Programm, das in einem guten Stil geschrieben ist und daher Bewunderung hervorruft. Allerdings kann dieser gute Stil mit Qualitätseinbußen hinsichtlich anderer Merkmale erkauft sein.



Anwenderprogramme

Solche Programme werden für spezielle Aufgaben geschrieben und bei Bedarf in den Speicher des Computers geladen. Im folgenden finden Sie einige Beispiele dafür.



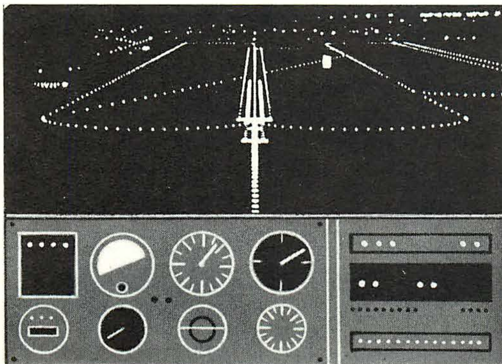


Datenbankprogramm: Ein Programm, das die Arbeit mit einer Sammlung von **Daten** ermöglicht. Dabei kann es sich um eigene Daten handeln, die man katalogisieren und auf dem neuesten Stand halten möchte, oder um bereits vorhandene Daten eines bestimmten Interessengebiets.

Expertensystem: Dies ist eine Art **Datenbank-Programm**, das eine große Anzahl ganz spezieller Daten enthält, z. B. aus den Bereichen Ingenieurwesen, Medizin oder Recht. Der Benutzer eines solchen Programms kann bestimmte Informationen eingeben, und der Computer greift auf seine Informationsspeicher zurück, um zu einer Entscheidung über ein Problem zu gelangen, für das man normalerweise einen Fachmann einschalten würde.

Textverarbeitungsprogramm: Mit diesem Programm kann man einzelne Textzeilen oder ganze Absätze auf dem Bildschirm abändern oder neu ordnen, indem man neue Zeilen hinzufügt und Fehler korrigiert. Dies ist sehr nützlich für alle, die längere Texte schreiben müssen.

Tabellenprogramm: Mit diesen Programmen kann man finanzielle Planungsdaten erstellen. Dazu baut man eine Datentabelle im Inneren des Computerspeichers auf. Man kann sich beliebige Teile der Tabelle auf den Bildschirm holen und die Daten, falls nötig, abändern. Wenn man einige Daten ändert, errechnet der Computer neue Ergebnisse, die von den geänderten Daten abhängen. Ein bekanntes Beispiel für ein solches Tabellenprogramm ist **VisiCalc**.



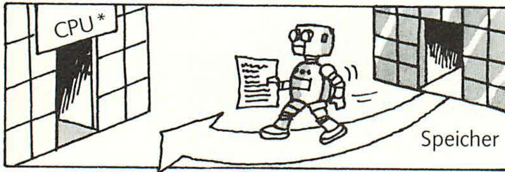


Simulationsprogramm: Diese Programme liefern ein Modell einer Situation, die in der Wirklichkeit vorkommt, z. B. das Flugzeugsimulationsprogramm in der Abbildung oben. Dieses gehört zu einem Spielprogramm; Flugsimulation wird jedoch auch in der Ausbildung von Piloten angewandt. Simulationen werden z. B. in der Industrie eingesetzt, um Produkte vor ihrer Massenherstellung zu testen. Ganz allgemein findet man sie im Geschäftsleben oder bei wirtschaftlichen Planungsvorhaben. Man kann z. B. den Aufbau einer Firma simulieren oder die voraussichtliche Entwicklung der Weltwirtschaft.

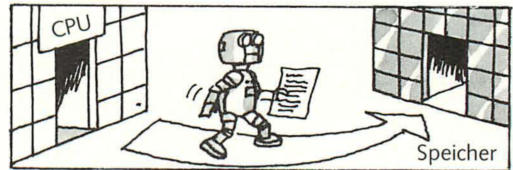
Abenteuerspiele: Für Abenteuerspiele braucht man einen großen Datenspeicher. Während des Spiels werden dem Spieler verschiedene Möglichkeiten eröffnet, um Gefahren zu entgehen; dabei muß er Fallen vermeiden und manchmal bestimmte Aufgaben lösen, um „durchzukommen". Um zu gewinnen, muß der Spieler alle Abenteuer bestehen und dabei noch möglichst viele Punkte sammeln.

Der Maschinencode

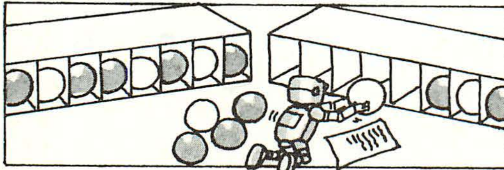
Ein **Maschinencode-Programm** ist entweder **hexadezimal** oder in **Assemblersprache** (siehe Seite 126/127) geschrieben. In einem Maschinencode-Programm richtet sich jeder Befehl direkt an den Inhalt eines **Speicherplatzes** oder eines **Registers** (das ist der Datenspeicher der **Zentraleinheit**). Die Menge der Maschinencode-Befehle, die ein Computer versteht, heißt **Befehlsvorrat**. Jeder Befehl löst eine Operation im Inneren des Computers aus. Im folgenden sind die Begriffe erklärt, mit denen diese Operationen beschrieben werden.



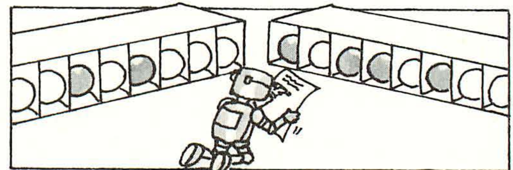
Laden: Informationen aus dem Speicher entnehmen und in ein **Register** übertragen.



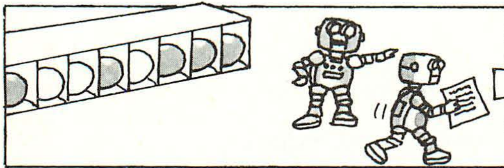
Speichern: Den Inhalt eines Registers in den Speicher übertragen.



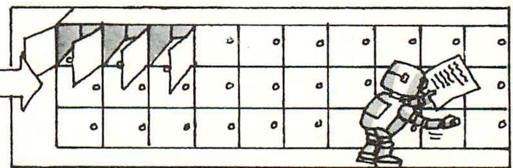
Übertragen: Die **Daten** eines Registers in ein anderes Register überführen.



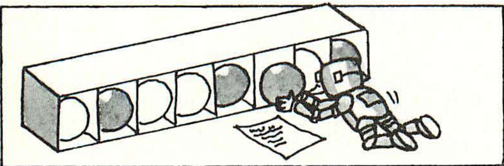
Vergleichen: Den Computer zum Vergleich zweier Daten veranlassen. Als nächster Befehl folgt normalerweise eine Verzweigung.



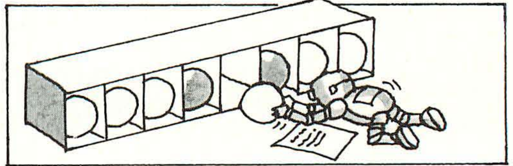
Sprung: Bewirkt, daß der Computer sich direkt an eine bestimmte **Speicheradresse** wendet oder eine bestimmte Anzahl von Speicherplätzen vor- oder zurückgeht, um zum nächsten Befehl zu gelangen.



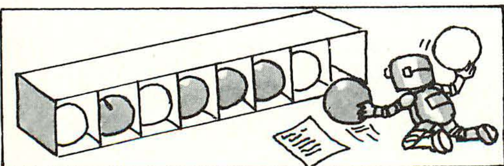
Verzweigung: Dies ist dasselbe wie ein **Sprung**, allerdings wird der Computer hier immer eine bestimmte Anzahl von Speicherplätzen vor- oder zurückgeschickt, um zur nächsten Anweisung zu gelangen.



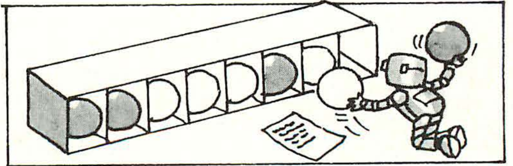
Setzen: Ein bestimmtes **Bit** ändern, d. h. das **Übertragskennzeichen** (siehe Seite 112) auf 1 setzen.



Löschen: Ein bestimmtes Bit auf 0 setzen oder es zurücksetzen.



Erhöhen: Zum Inhalt eines Registers oder Speicherplatzes 1 hinzufügen.

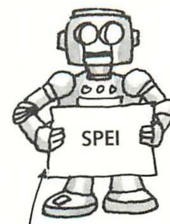


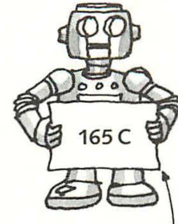
Vermindern: Den Inhalt eines Registers oder Speicherplatzes um 1 vermindern.

*) CPU = Zentraleinheit (englische Abkürzung für **C**entral **P**rocessing **U**nit).

Adressierungsarten

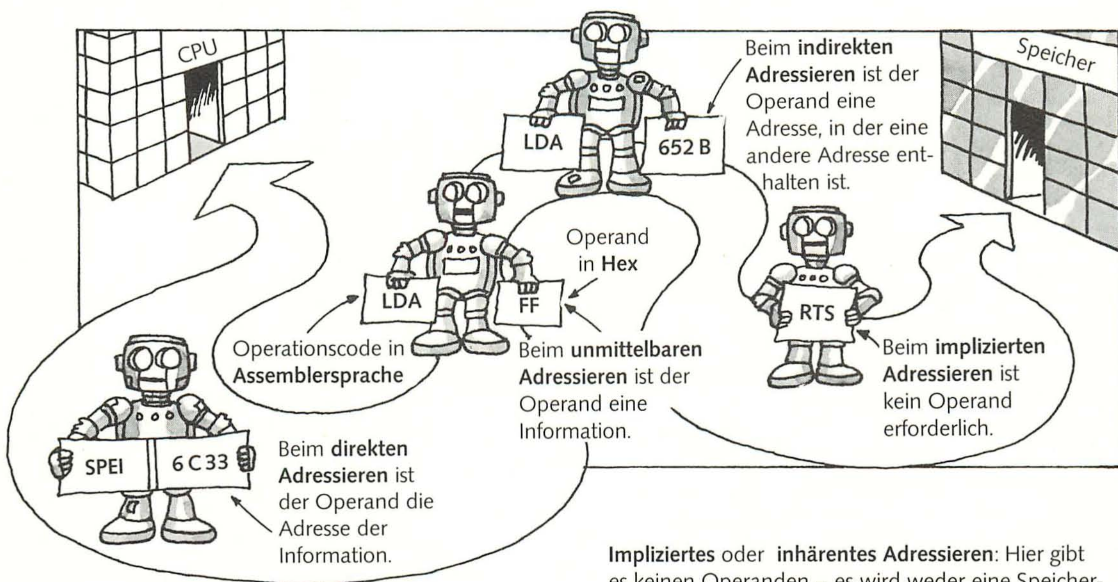
Adressierungsarten sind die verschiedenen Möglichkeiten, dem Computer zu sagen, wo er die nächste Information findet. Jede **Maschinencode**-Programmzeile besteht aus zwei Teilen: dem **Operationscode**, also der Anweisung, und dem **Operanden**, der dem Computer sagt, wo er die **Daten** findet, die er bearbeiten soll. Einige der am häufigsten vorkommenden Adressierungsarten sind im folgenden dargestellt.





Unmittelbare Adressierung: In dieser Betriebsart ist der Operand selbst die Information. Der Operationscode enthält oft den Namen eines Registers. Diese Adressierungsart heißt auch **explizites Adressieren**.

Direktes oder **absolutes Adressieren**: In dieser Betriebsart ist der Operand die Adresse, unter der die Information gefunden werden kann, bekannt als direkte oder absolute Adresse.



Indirektes Adressieren: Hier ist der Operand die Adresse eines Speicherplatzes oder Registers, in dem die Adresse der Information gespeichert ist. Die im Operand angegebene Adresse heißt **Ablaufadresse**, **Dispatch-Adresse** oder **Sprungvektor**.

Indiziertes Adressieren: In dieser Betriebsart ist der Operand lediglich ein Teil der benötigten Adresse. Zur Vervollständigung muß der Inhalt eines Registers, des **Indexregisters**, hinzugefügt werden. Die vom Operanden festgestellte Adresse heißt **Basisadresse**, die Zahl im Indexregister ist der **Umstellungswert**, und die endgültige Adresse, die durch Addition der beiden zustande kommt, ist die **effektive Adresse**. Diese Adressierungsart wird oft in Verbindung mit einer anderen verwendet – so gibt es z. B. auch **indirektes indiziertes Adressieren**.

Impliziertes oder **inhärentes Adressieren**: Hier gibt es keinen Operanden – es wird weder eine Speicheradresse noch ein Register festgesetzt. Der Computer bekommt die Information, was er zu tun hat, vom Operationscode selbst.

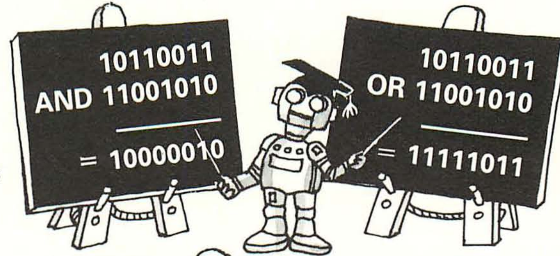
Relatives Adressieren: Bei dieser Adressierungsart wird der Computer angewiesen, eine bestimmte Anzahl von Speicherplätzen voraus- oder zurückzugehen, ausgehend von dem Speicherplatz, mit dem er sich gerade befaßt.

Null-Seiten-Adressieren: Hier ist der Operand eine Adresse auf der ersten Seite des Speichers, die man **Null-Seite** nennt. Gewöhnlich enthält jede Seite des Speichers 256 **Speicherplätze** (siehe Seite 115). Die Null-Seite enthält die ersten 256, die von 0 bis 255 numeriert sind. Die meisten Adressen beanspruchen zwei **Bytes** des Speichers. Die Adressen auf der Null-Seite können jedoch an einem Speicherplatz gehalten werden, d. h. sie können als Binärzahl mit acht Ziffern dargestellt werden. Beim Null-Seiten-Adressieren wird für die Adresse also nur ein Byte benötigt. Daher ist diese Adressierungsart schneller als andere.

Der Maschinencode (Fortsetzung)

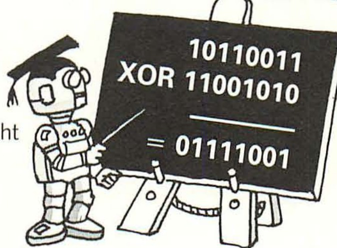
Manche Maschinencode-Befehle ermöglichen es, innerhalb eines **Bytes** ein einziges **Bit** abzuändern. Einige dieser Befehle werden für Rechenaufgaben benutzt, andere dienen dazu, bestimmte **Zeichen** auf dem Bildschirm zu bewegen. Im folgenden werden einige der am meisten vorkommenden Operationen erklärt, die man mit diesen Befehlen ausführen kann.

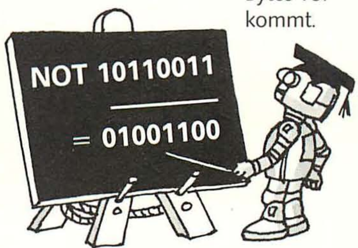
AND ▶ Bei dieser Operation vergleicht der Computer den **Binärcode** für zwei Informationen. Das Ergebnis ist 1, wenn in beiden Bytes eine 1 an entsprechender Stelle steht. Anderenfalls ist das Ergebnis 0.

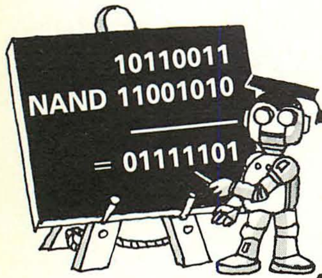


OR: Diese Operation erzeugt eine 1, wenn eine 1 entweder an einer oder an beiden Stellen der beiden Bytes vorkommt.

XOR ▶ Darunter versteht man das ausschließende **OR**. Es erzeugt eine 1, wenn eine der beiden Bits gleich 1 ist, aber nicht beide.

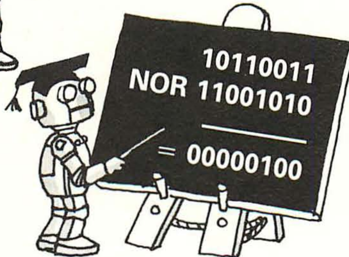






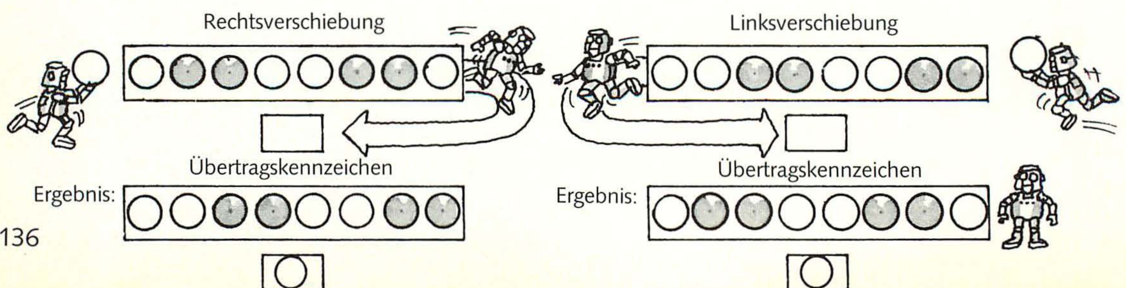
◀ **NAND**: Bedeutet **NOT AND** und erzeugt das **Komplement** (siehe **NOT**) der Zahl, die entsteht, indem zwei Bytes durch **AND** verbunden werden.

NOT: Unter dieser Operation versteht man die **Bit-Komplementbildung**, d. h. alle Einsen in einem Byte werden zu Nullen gemacht und umgekehrt. Das Ergebnis einer NOT-Operation ist das **Komplement** (manchmal auch **logisches Komplement**) der ursprünglichen Zahl. Die Summe der ursprünglichen Zahl und ihres Komplements ergibt immer 11111111 (dezimal 255). Diese Tatsache ist Grundlage einer Vielzahl mathematischer Operationen.

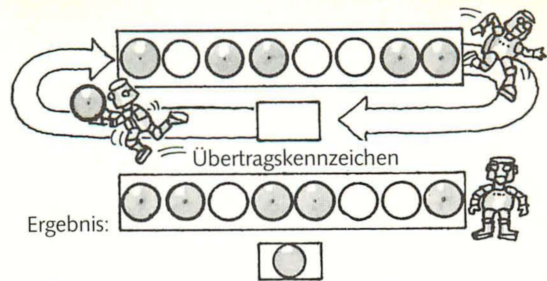


NOR: Bedeutet NOT OR und erzeugt das **Komplement** (siehe **NOT**) eines **OR**-Ergebnisses.

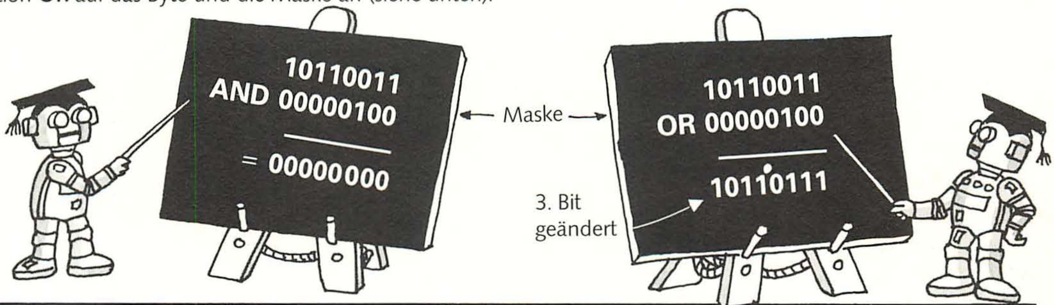
Verschieben: Diese Operation „verschiebt" die Bits eines Bytes um einen Platz nach rechts oder nach links. Das Bit, das am Ende „herausfällt", kann das **Übertragskennzeichen** oder **Carry Flag** (siehe Seite 112) ergeben. Am anderen Ende wird entweder eine 1 oder eine 0 angefügt, je nach der angewandten Verschiebeart. Es gibt verschiedene Arten der Verschiebung, die für verschiedene Zwecke angewendet werden. So kann z. B. eine Verschiebung nach rechts dazu dienen, eine Zahl zu halbieren, wie im Beispiel unten. Hier ist aus der Binärzahl für 102 (dezimal) die Binärzahl 51 geworden. Eine Linksverschiebung kann man zur Verdoppelung einer Zahl verwenden. Das Beispiel unten zeigt, wie die Binärzahl für 51 in 102 verdoppelt wird.



Rotieren: Funktioniert wie das **Verschieben**, allerdings wird das **Bit** am einen Ende des Bytes oder dasjenige, das sich bereits im **Übertragskennzeichen** oder **Carry Flag** befindet (siehe Seite 112), zum anderen Ende des **Bytes** addiert.



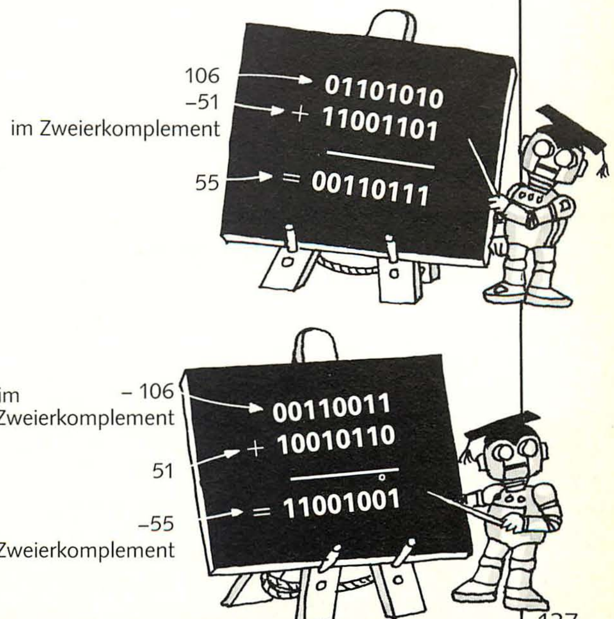
Maske: Möchte man wissen, ob ein bestimmtes Bit in einem Byte eine 1 oder eine 0 ist, dann stellt man ein Byte her, das man Maske nennt. Diese Maske hat sieben Nullen und eine Eins. Die Eins befindet sich dabei an der Stelle des Bits, um das es einem geht. Wird eine **AND**-Operation zwischen der Maske und dem zu testenden Byte durchgeführt, dann ist das Resultat entweder 00000000 oder es stimmt mit der Maske überein.
Will man z. B. überprüfen, ob das dritte Bit eines Bytes eine 1 ist, dann wendet man die AND-Operation auf das Byte und 00000100 an (siehe unten). Ist das dritte Bit des getesteten Bytes eine 1, so ist das Resultat 00000100. Ist das dritte Bit eine 0, dann erhält man 00000000. Diesen Vorgang nennt man **Maskieren**.
Will man das Bit in 0 ändern, dann wendet man die Operation AND auf das Byte und auf das **Komplement** der Maske (siehe **NOT**) an. War das Bit 0 und möchte man es in 1 verändern, dann wendet man die Operation **OR** auf das Byte und die Maske an (siehe unten).



Negative Zahlen

Binär mit Vorzeichen: Dies ist einer der beiden wichtigsten Arten, in der die **Zentraleinheit** negative Zahlen verarbeiten kann. Dieses Verfahren wird heute nicht mehr allzuhäufig verwendet, da es durch die **Zweierkomplement-Notierung** (siehe unten) ersetzt worden ist. Im Vorzeichen-Binärcode ist das übrigbleibende Bit eines Bytes das **Vorzeichenbit**. Dabei ist die 0 für eine negative Zahl und die 1 für eine positive Zahl vorgesehen. Die Subtraktion erfolgt genauso wie sonst.

Zweierkomplement-Notierung: Die meisten Computer verarbeiten negative Zahlen heute auf diese Weise. Das Zweierkomplement einer Zahl wird verwendet als ihr negativer Gegenwert und gebildet durch das Komplement der Zahl (siehe **NOT**) unter nachfolgender Addition von 1. So ergibt die **Binärzahl** tatsächlich 00000000, wenn sie zur ursprünglichen Zahl addiert wird (mit einer „mitgenommenen" 1). Das Zweierkomplement von 00110011 (dezimal 51) ist z. B. 11001101. Dies wird zur Darstellung von 51 als Dezimalzahl benutzt.
Bei der Zweierkomplement-Notierung subtrahiert der Computer eine Zahl, indem er ihr Zweierkomplement zu der Zahl addiert, von der es subtrahiert werden soll (siehe rechts). Tatsächlich führt die Berechnung der folgenden Dezimalsummen zum gleichen Ergebnis: 106 – 51 ist das gleiche wie 106 + (–51). Ergibt eine Rechnung ein negatives Ergebnis, z. B. 51 – 106, dann erscheint dies in der Zweierkomplement-Form (siehe rechts).



Mathematische Begriffe und Zahlensysteme

Argument: Bei einer mathematischen Funktion, z. B. bei der Quadratwurzel, versteht man unter dem Argument die Zahl, mit der die Funktion arbeitet. Beispiel: Der **BASIC**-Befehl zur Berechnung der Quadratzahl von 9 lautet SQR(9); dabei ist die Zahl 9 das Argument.

Exponent: Dies ist das Wort für die Potenz, in die man eine Zahl erhebt. In der Aussage $X = 2^8$ ist 8 der Exponent.

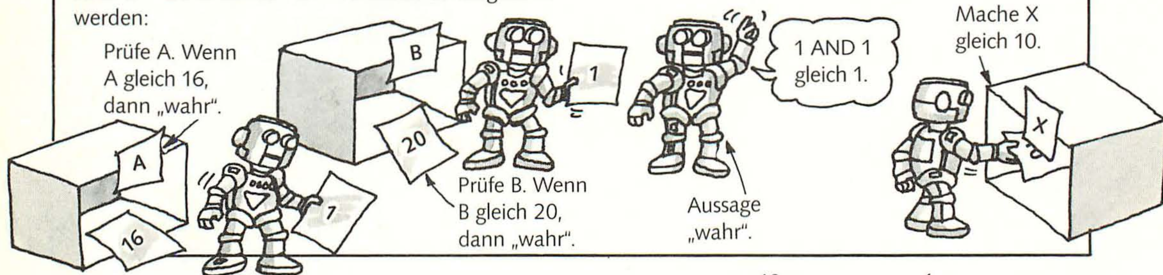
Mantisse: Sehr große Zahlen werden oft dargestellt als eine Zahl, die mit einer bestimmten Potenz von 10 zu multiplizieren ist. So bedeutet z. B. $64 \cdot 10^6$ die Zahl 64 Millionen. Die Zahl 64 ist dabei die Mantisse.

Boole'sche Algebra

Dieses System wurde von George Boole, einem irischen Mathematiker, im 19. Jahrhundert erfunden. Es stellt die Grundlage für die gesamte Computerlogik dar. Dabei handelt es sich um eine Anzahl von Regeln, die darauf beruhen, daß Aussagen entweder wahr oder falsch sind. Mit Hilfe dieser Regeln kann man **Wahrheitstabellen** aufstellen (siehe das Beispiel rechts). Solche Tabellen gibt es für die Operationen **AND**, **OR** und **NOT** (siehe Seite 136). Mit Hilfe dieser Regeln können logisch begründete Rechnungen durchgeführt werden. Beispiel: Der Befehl IF A = 16 AND B = 20 THEN LET X = 10 würde so ausgeführt werden:

AND	Wahrheitstabelle		
1	AND	1	1
1	AND	0	0
0	AND	1	0
0	AND	0	0

In dieser Tabelle bedeutet 1 „wahr" und 0 „falsch".



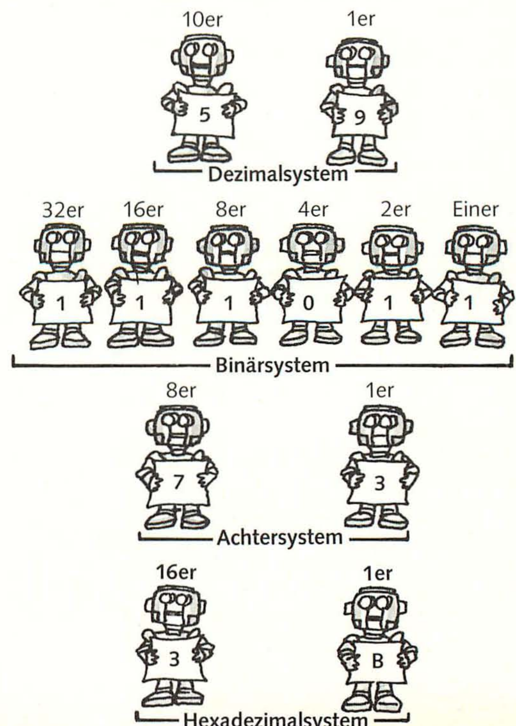
Zahlensysteme

Jede Zahl hat ihre **Basiszahl**. Die Basiszahl unseres **Dezimalsystems** ist z. B. 10. Das Dezimalsystem verwendet 10 Ziffern (0 bis 9). In einer Zahl steht jede Ziffer in einer bestimmten Spalte, deren Wert 10mal so groß ist wie der Wert der Spalte rechts davon, d. h. es gibt Einer, Zehner, Hunderter usw.

Binärsystem: Dieses System kennt nur zwei Ziffern, nämlich 0 und 1. Der Wert jeder Ziffer in einer Zahl ist also doppelt so hoch wie der Wert in der Spalte rechts davon; es gibt also Einer, Zweier, Vierer, Achter usw.

Achtersystem: Dieses System besteht aus acht Ziffern (0 bis 7). Jede Ziffer in einer Zahl besitzt einen Wert, der achtmal so groß ist wie der Wert in der Spalte rechts davon; es gibt also Einer, Achter, Vierundsechziger usw.

Hex (Abkürzung für **Hexadezimal**): In diesem System werden 16 Zeichen verwendet, nämlich die Ziffern 0 bis 9 und die Buchstaben A bis F. Der Wert jeder Hex-Spalte ist 16mal so hoch wie der Wert in der Spalte rechts davon.



Chips

Halbleiter: Darunter versteht man Stoffe, die Elektrizität nur unter bestimmten Umständen leiten, z. B. dann, wenn die Temperatur auf eine bestimmte Höhe gebracht wird oder wenn sie mit Chemikalien behandelt werden. Chips werden aus einem Halbleiter namens **Silizium** hergestellt und mit Chemikalien behandelt, so daß nur bestimmte Bahnen auf dem Chip Elektrizität leiten. Diese Behandlung heißt **Doping**.

N-Silizium: Dieses Halbleitermaterial entsteht, wenn Silizium mit Phosphor behandelt wird. Die behandelten Bereiche lassen nur negativ geladene Teilchen durch.

P-Silizium: Dieses Halbleitermaterial entsteht, wenn Silizium mit Bor behandelt wird. In so behandelten Bereichen des Chips fehlen negativ geladene Teilchen. Diese Fehlstellen („Löcher") können sich wie positiv geladene Teilchen bewegen.

MOS (**M**etal **O**xide **S**emiconductor, d. h. Metalloxid-Halbleiter): Bei diesen Chips dient Metall als elektrischer Leiter und Siliziumoxid als Isolierschicht. Andere Arten von Halbleiter-Schaltkreisen auf Chips sind **bipolare** und **SOS-Chips** (Silizium auf Saphir).

NMOS- und **PMOS-Chips**: Dabei handelt es sich um verschiedene Arten von MOS-Chips, für die entweder nur **N-** oder nur **P-Silizium** verwendet wird. PMOS-Chips haben im allgemeinen mehr Schaltkreise, während NMOS-Chips viel schneller arbeiten.

CMOS-Chips: Ein CMOS-Chip besteht aus **N-** und **P-Silizium**. Dieser Chiptyp ist meist nicht so schnell wie **NMOS-Chips**, er benötigt jedoch weniger Strom und wird daher oft in Taschencomputern eingebaut.

DIP (**D**ual **I**n-line **P**ackage): Darunter versteht man einen Chip im Endzustand, also in einen Rahmen eingesetzt und in ein Kunststoffgehäuse eingebaut, mit zwei parallelen Reihen von „Beinen" oder **Pins**. Diese Pins transportieren die elektrischen Signale vom und zum Chip.

Eingabe-/Ausgabechips

Im folgenden sind die gängigsten Chiptypen aufgeführt, die in **Schnittstellen** von Computern verwendet werden:

UART (**U**niversal **A**synchronous **R**eceiver/**T**ransmitter): Dient zur Umwandlung von seriellen Signalen in parallele Signale und umgekehrt.

ACIA (**A**synchronous **C**ommunication **I**nterface **A**daptor): Dient ebenfalls zur Umwandlung von seriellen in parallele Signale.

PIO (**P**arallel **I**nput/**O**utput): Wird bei der Verbindung des Computers mit einem **Peripheriegerät** verwendet, das ebenfalls parallel arbeitet.

PIA (**P**eripheral **I**nterface **A**daptor): Arbeitet genauso wie ein **PIO**-Chip.

Speicherchips

PROM (**P**rogrammable **ROM**): Eine besondere Art von **ROM-Chip**, in den sich nach der Herstellung Programme einschreiben lassen, indem man diese mit Hilfe eines speziellen Geräts in den Chip „einbrennt". Die Programme sind dann fest im Chip gespeichert.

EPROM (**E**rasable **P**rogrammable **ROM**): Im Prinzip ein **PROM**-Chip, dessen Programm man jedoch löschen und durch ein neues ersetzen kann. Zum Löschen ist jedoch ein besonderes Gerät nötig, das ultraviolettes Licht erzeugt.

EEPROM (**E**lectrically **E**rasable **P**rogrammable **ROM**), auch noch bekannt unter der Bezeichnung EAPROM/EAROM Chip (**E**lectrically **A**lterable): Dieser Chip ist vergleichbar mit dem **EPROM**-Chip, dessen Programm jedoch durch elektrische Signale gelöscht und neu eingeschrieben werden kann.

Integrationsdichte

Darunter versteht man die Anzahl der **Gatter** (siehe Seite 112), die auf ein Chip passen. Es gibt folgende Einteilung:

SSI (**S**mall **S**cale **I**ntegration): Enthält weniger als 10 Gatter.

MSI (**M**edium **S**cale **I**ntegration): Enthält 10 bis 100 Gatter.

LSI (**L**arge **S**cale **I**ntegration): Enthält 100 bis 10 000 Gatter.

VLSI (**V**ery **L**arge **S**cale **I**ntegration): Enthält über 10 000 Gatter.

Register

Dieser Ausgabe liegen folgende englische Titel zugrunde:
Understanding the Micro, Introduction to Computer Programming, Usborne Guide to Computer Jargon
Aus dem Englischen übersetzt von Dipl.-Math. Barbara Schumacher und bearbeitet von Hubert Hofrat und Martin Stübs
Redaktion der englischen Originalausgaben: Lisa Watts
Redaktion der deutschen Ausgabe: Micha Ramm
Illustrationen: Jeremy Banks, Kuo Kang Chen, Tim Cowdell, Mark Longworth, Chris Lyon, Martin Newton, Graham Round, Martin Salisbury, Graham Smith und Ian Stephen
Buchgestaltung: Kim Blundell, Roger Priddy und Graham Round
Umschlaggestaltung: Erich Gebhardt.

Lizenzausgabe mit Genehmigung des Otto Maier Verlages, Ravensburg, für die Bertelsmann Club GmbH, Gütersloh, die Europäische Bildungsgemeinschaft Verlags-GmbH, Stuttgart, die Buchgemeinschaft Donauland, Kremayr & Scheriau, Wien, und die Buch- und Schallplattenfreunde GmbH, Zug/Schweiz.
Diese Lizenz gilt auch für die Deutsche Buchgemeinschaft C.A. Koch's Verlag Nachf., Berlin-Darmstadt-Wien.

Titel-Nr.: 04705 0

Computer sind kein Hexenwerk.
Sie gehören inzwischen zum Alltag in Betrieben, in Schulen und für viele auch in der Freizeit.
Sie müssen nicht Experte sein oder sich erst mühsam spezielle Fachkenntnisse erwerben, wenn Sie diesem elektronischen Hobby nachgehen wollen.
Dieser reich illustrierte Band gibt Ihnen in leichtverständlicher Sprache einen Einblick in Technik und Funktionsweisen und zeigt Ihnen die vielfältigen Anwendungsmöglichkeiten von Heimcomputern.
Damit können Sie Ihr Interesse an moderner Technik aktiv nutzen – und Sie werden sehen, daß das auch noch Spaß macht!